2024 International Conference on Computing, Networking and Communications (ICNC)

Graph Isomorphism—*Characterization and Efficient Algorithms*

Jian Ren and Tongtong Li
Department of ECE, Michigan State University, East Lansing, MI 48824 USA. email: {renjian, tongli}@msu.edu

***Abstract*—The graph isomorphism problem involves determining whether two graphs are isomorphic and the computational complexity required for this determination. In general, the problem is not known to be solvable in polynomial time, nor to be NP-complete. In this paper, we prove that for undirected graphs, the complexity in determining whether two graphs are isomorphic is at most $\mathcal{O}(n^3)$.**

***Index Terms*—Undirected graph, characterization, isomorphism, algorithm, polynomial time complexity**

I. Introduction

Graphs are data structures used to represent objects and their relationships [1]. The objects are also sometimes referred to as nodes or vertices, while the relationships are known as edges. Essentially, graphs provide descriptions of items that are interconnected by relations.

Graphs are widely used in machine learning as a tool to predict links and classify nodes [2]. By loading the data into the graph database, the data science library can be used to train a machine learning model and make predictions.

Deciding whether two graphs are isomorphic is a classical algorithmic problem that has been researched since the early days of computing. Graph isomorphism involves determining when two graphs possess the same data structures and data connections [3]. It is widely used in various areas such as social networks, computer information system, image processing, protein structure, chemical bond structure, etc.

Unfortunately, the general graph isomorphism problem is not known to be solvable in polynomial time nor to be NP-complete, and therefore may be in the computational complexity class NP-intermediate [4], [5]. As a result, this problem was viewed as an open problem [6], [7].

In this paper, we investigate the graph isomorphism problem using the eigenvalues and eigenvectors of the adjacency matrices of the graphs. Eigenvalues and eigenvectors of square matrices have found extensive applications across various domains. Eigenvalues are used in computer graphics to perform transformations on objects, such as rotating or scaling. For example, when an image is resized, the eigenvalues of its covariance matrix can be used to preserve its principal components and avoid distortion, because the eigenvectors of the covariance matrix are actually the directions of the axes, while eigenvalues are simply the coefficients attached to eigenvectors, which given the amount of variance carried in each principal component [8]. Eigenvalues have been widely used in signal processing to extract meaningful features from large datasets. For example, in image processing, the eigenvalues of a matrix of pixel intensities can be used to identify the most significant patterns and structures in the image [9]. Google's extraordinary success as a search engine was due to their clever use of eigenvalues and eigenvectors [10]. Claude Shannon utilized eigenvalues to calculate the theoretical limit of channel capacity. The eigenvalues are then essentially the gains of the channel's fundamental modes, which are recorded by the eigenvectors. Eigenvalues have also been employed to analyze the stability of structures and machines, such as determining the natural frequency of a bridge and assessing the likelihood of bridge oscillations or even collapse under specific conditions.

The rest of this paper is organized as follows: In Section II, the preliminary is provided. Our main results are presented in Section III. We conclude in Section IV.

II. Preliminary

A. Undirected Graph and Adjacency Matrix

An undirected graph is generally represented as a pair $G = (V, E)$, where V is the set of vertices, and $E \subset V \times V$ is the set of edges satisfying $(u, v) \in E$ if and only if $(v, u) \in E$. The neighbors of a vertex v is $N(v) = \{w : (v, w) \in E\}$.

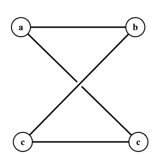
In graph theory, we say that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection f between the vertex sets of V_1 and V_2 $f : V_1 \to V_2$, such that any two vertices u and v of G_1 are connected in G_1 if and only if $f(u)$ and $f(v)$ are connected in G_2, i.e., $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$. If an isomorphism exists between two graphs, then the graphs are called isomorphic and denoted as $G_1 \simeq G_2$.

In graph theory, the degree of a vertex v is the number of edges connecting it, called degree of the vertex v and denoted as $\deg(v)$. It is obvious that $\deg(v) = |N(v)|$. From the definition of isomorphism, $G_1 \simeq G_2$ implies that $\deg(v) = \deg(f(v))$, which also implies that if $\deg(v) \neq \deg(f(v))$, then we cannot match up the two vertices.

In many applications, each edge E of a graph is associated with a numerical value called a weight, denoted as $w(E)$, which might represent for example costs, lengths or capacities, depending on the problem at hand. In this paper, we consider the weight of all edges to be 1.

For a graph with vertex set $V = \{v_1, \cdots, v_n\}$, the adjacency matrix, sometimes also called the connection matrix, is a square $n \times n$ $(0, 1)$-matrix A such that its element $A_{ij} = A_{ji} = 1$ if there is an edge from vertex v_i to vertex v_j, and 0 if there is no edge, and also $A_{ii} = 1$ for all i, that is 1's on its diagonal elements [11]. The elements of the matrix indicate whether pairs of vertices are adjacent or connected in the graph. If the graph is undirected (i.e., all of its edges are bidirectional), the adjacency matrix is symmetric, that is $A_{ij} = A_{ji}$.

Example 1. *For the graph given below, the corresponding adjacency matrix is shown to the right.*



$$\Longrightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Definition 1. *Let A be an adjacency matrix of a graph G. We represent the matrix obtained by interchanging the i^{th} and the j^{th} rows and the i^{th} and the j^{th} columns of matrix A as $A[i \leftrightarrow j]$. We will refer to this operation as the (i, j) interchanging of matrix A for simplicity.*

The $A[i \leftrightarrow j]$ operation defined in Definition 1 can be represented in matrix multiplication form as follows:

$$A[i \leftrightarrow j] = E_{ij} A E_{ij}^T,$$

where E_{ij} is the matrix derived by interchanging the i^{th} and j^{th} rows of the identity matrix I_n, that is

$$E_{ij} = \begin{array}{c} \begin{array}{ccccccc} 1 & & i & & j & & n \end{array} \\ \begin{bmatrix} 1 &&&&&& \\ & \ddots &&&&& \\ && 0 & \cdots & 1 && \\ && \vdots & \ddots & \vdots && \\ && 1 & \cdots & 0 && \\ &&&&& \ddots & \\ &&&&&& 1 \end{bmatrix} \end{array} \begin{array}{c} 1 \\ \\ i \\ \\ j \\ \\ n \end{array}.$$

The number of 1's in a row or a column of matrix A is referred as the *weight* of the row or column.

B. Eigenvalues and Eigenvectors

For an $n \times n$ square matrix A, a scale λ is called an *eigenvalue* [12] if there exists a vector $\boldsymbol{u}$ such that

$$A\boldsymbol{u} = \lambda \boldsymbol{u}. \tag{1}$$

In this case, $\boldsymbol{u}$ is called an *eigenvector* of matrix A associated with eigenvalue λ.

Let A be an $n \times n$ matrix, then the expression

$$\det(xI - A) \tag{2}$$

is a polynomial, called the *characteristic polynomial* of matrix A, and

$$\det(xI - A) = 0 \tag{3}$$

is called the *characteristic equation*. The eigenvalues λ's of A defined in equation (1) are solutions of the characteristic equation (3).

It follows from equation (3) that if λ is an eigenvalue of A, then there exists a nonzero eigenvector $\boldsymbol{u}$ for equation (1).

For an $n \times n$ matrix A with characteristic polynomial given by equation (2), the *multiplicity* of an eigenvalue λ of A is the number of times λ occurs as a root of that characteristic polynomial equation (3).

If A is a real symmetric matrix, then its eigenvalues are all real numbers and the eigenvectors corresponding to distinct eigenvalues are orthogonal. If A is a real $n \times n$ symmetric matrix, then there exists an orthonormal (orthogonal and unit vector) set of eigenvectors that forms the basis of the n dimensional vector space.

III. Our Main Results

Theorem 1. *The interchange operations on adjacency matrices will not alter the weight of the columns or rows of the matrix.*

Proof. Let A be an $n \times n$ adjacency matrix of a graph and $1 \le i, j \le n$ are two integers, $i \ne j$. The matrix $A[i \leftrightarrow j]$ is derived from matrix A by interchanging the i^{th} and j^{th} rows and columns of A, resulting in the interchanging A_{ii} with A_{jj}, and A_{ij} with A_{ji}. Since A is a symmetric matrix with diagonal elements equal to 1, these four elements in A remain unchanged. Therefore, the weight of the i^{th} row or column is simply exchanged with the j^{th} row or column, while the overall weight of the matrix A remains unchanged. □

Note that this theorem holds due to the special structure of the adjacency matrix, and it does not hold true in general even for symmetric matrices.

Theorem 2. *Let A_1 and A_2 be the adjacency matrices of graphs G_1 and G_2, respectively. Then G_1 and G_2 are isomorphic if and only if there exists a sequence of interchange operations (i.e., a permutation matrix) that transforms the adjacency matrix A_1 to A_2.*

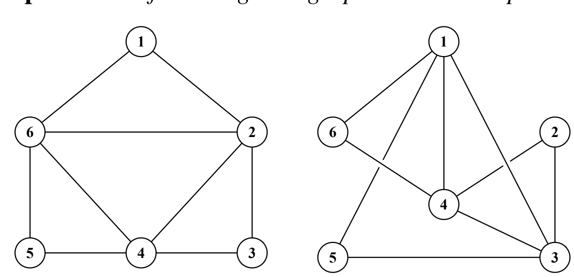
The proof of this theorem can be found in the full paper.

In addition to bridge isomorphism and interchange operations, Theorem 2 also provides an efficient algorithm to transform graph G_1 into its isomorphic counterpart G_2, as presented in Algorithm 1.

Algorithm 1 Transform graph G_1 to its isomorphic graph G_2 by transforming adjacency matrices A_1 to A_2

Let $V_1 = \{1, 2, \cdots, n\}, V_2 = \{f(1), f(2), \cdots, f(n)\}$
for $i = 1$ to n **do**
  $t = f(i)$
  for $j = 1$ to n **do**
    while $t < i$ **do**
      $t = f(t)$
    end while
  end for
  $g(i) = t$
  $A_1 = A_1[i, g(i)]$
end for

Example 2. *The following two graphs are isomorphic.*

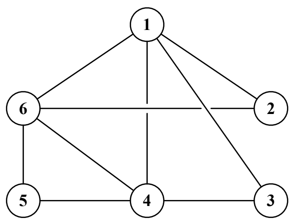


The adjacency matrices of the two graphs A and B are given below:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

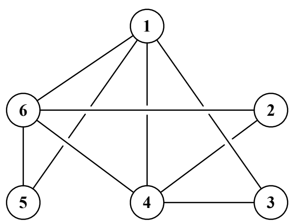
Suppose we want to transfer the vertices of V_1 to $V_2 = \{2, 5, 3, 4, 1, 6\}$. Based on Algorithm 1, we can transform the graph from matrix A to the graph in matrix B through the following sequence of interchanging operations.

1) $A[1 \leftrightarrow 2]$, *which transforms the graph with adjacency matrix A and the following graph and adjacency matrix A_1:*



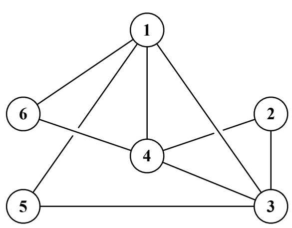
$$A_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

2) $A[2 \leftrightarrow 5]$, *which further transforms the graph with adjacency matrix A_1 and the following graph and adjacency matrix A_2:*



$$A_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

3) $A[3 \leftrightarrow 6]$, *which finally transforms the graph and the corresponding adjacency matrix from A_2 to B:*



$$B = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Example 3. *Based on Algorithm 2, we can transform matrix B to matrix A, which transform the vertices $V_2 = \{1, 2, 3, 4, 5, 6\}$ to $V_1 = \{5, 1, 6, 4, 2, 3\}$, through the following interchanging operations:*

1) $B[2 \leftrightarrow 1]$, *which transforms B to*

$$B_1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Algorithm 2 Transform graph G_2 back to its isomorphic graph G_1 by transforming adjacency matrices A_2 to A_1

Let $V_1 = \{1, 2, \cdots, n\}, V_2 = \{f(1), f(2), \cdots, f(n)\}$
for $i = 1$ to n **do**
    $t = f(i)$
    for $j = 1$ to n **do**
        while $t > i$ **do**
            $t = f(t)$
        end while
    end for
    $h(i) = t$
    $A_2 1 = A_2[i, h(i)]$
end for

---

2) $B[5 \leftrightarrow 1]$, *which transforms B_1 to*

$$B_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

3) $B[6 \leftrightarrow 3]$, *which transforms B_2 to A.*

Since the interchange is an elementary matrix operation, it does not alter the eigenvalues of the adjacency matrix. Therefore, we have the following corollaries.

Corollary 1. *Let A_1 and A_2 be the adjacency matrices of graphs G_1 and G_2, respectively. If G_1 and G_2 are isomorphic, then their eigenvalues are the same.*

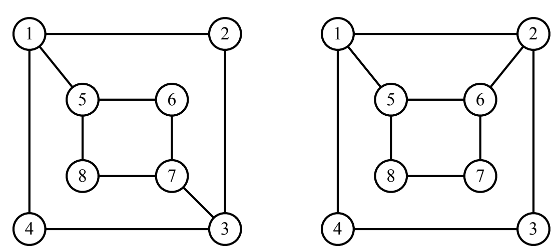
Corollary 2. *Let A_1 and A_2 be the adjacency matrices of graphs G_1 and G_2, respectively. If the eigenvalues of A_1 and A_2 are different, then they are not isomorphic.*

Corollary 3. *Let A and B be $n \times n$ adjacency matrices of two graphs. If the two graphs are isomorphic, then the total number of 1's (corresponding to the edges in the graphs) in the two matrices should be the same.*

From Corollary 3, we can conclude that if the number of 1's of two matrices are different, then the two graphs are not isomorphic.

However, the inverse of Corollary 3 is not true. In other words, even if two matrices have the same number of 1's, they may not be isomorphic.

Example 4. *For the following two graphs,*



their adjacency matrices are

$$G = \begin{bmatrix} 1&1&0&1&1&0&0&0\\ 1&1&1&0&0&0&0&0\\ 0&1&1&1&0&0&1&0\\ 1&0&1&1&0&0&0&0\\ 1&0&0&0&1&1&0&1\\ 0&0&0&0&1&1&1&0\\ 0&0&1&0&0&1&1&1\\ 0&0&0&0&1&0&1&1 \end{bmatrix}, \; H = \begin{bmatrix} 1&1&0&1&1&0&0&0\\ 1&1&1&0&0&1&0&0\\ 0&1&1&1&0&0&0&0\\ 1&0&1&1&0&0&0&0\\ 1&0&0&0&1&1&0&1\\ 0&1&0&0&1&1&1&0\\ 0&0&0&0&0&1&1&1\\ 0&0&0&0&1&0&1&1 \end{bmatrix}.$$

Matrices G and H contain the same number of 1's. However, due to 0 being an eigenvalue of G but not of H, the two corresponding graphs are not isomorphic.

Next, suppose two graphs are isomorphic, based on the definition of graph isomorphism, the corresponding vertices should have the same degree. Moreover, the subsequent vertex should also have the same degree and structure. The vertex and the associated structure are refereed to as the *vertex tree*. Therefore, the corresponding vertices should have the same vertex trees.

Based on this discovery, we can derive Algorithm 3.

Algorithm 3 Derive an isomorphic function to transform graph G_1 to graph G_2

Let G_1 and G_2 be two graphs and their vertex sets are $V_1 = \{v_1, \cdots, v_n\}$ and $V_2 = \{\bar{v}_1, \cdots, \bar{v}_n\}$, respectively.
Derive the degree tree of all the vertex of both graph G_1 and graph G_2.
repeat
  Select a vertex $v \in V_1$.
  if no vertex in V_2 has the same vertex tree as v **then**
    $G_1 \not\cong G_2$ and **stop**
  else
    Select a vertex $\bar{v} \in V_2$, that has the same vertex tree as v and map $f \colon v \to \bar{v}$.
    $V_1 \leftarrow V_1 \backslash \{v\}$ and $V_2 \leftarrow V_2 \backslash \{\bar{v}\}$.
  end if
until $V_1 = \emptyset$, or no $\bar{v} \in V_2$ for the selected v.

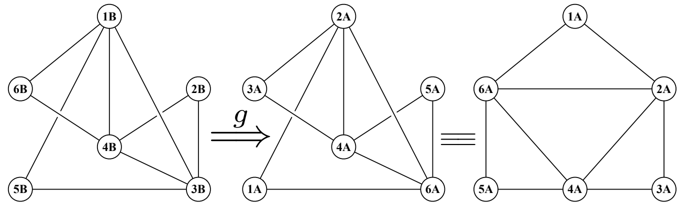
Example 5. *For the two graphs given in Example 2, the degrees for the 6 vertices of graphs A and B are given below:*

$$A \colon \{2, 4, 2, 4, 2, 4\}, \;\; B \colon \{4, 2, 4, 4, 2, 2\}.$$

Based on the degree information, we can derive the following mapping g, which transforms graph B to graph A:

1) *Since the degree of node 1_B is 4, it can only be mapped to one of the nodes in $\{2_A, 4_A, 6_A\}$. Let's map node 1_B to node 2_A, i.e., define $g(1_B) = 2_A$.*
2) *Node 2_B has degree 2, so it can only be mapped to a node in $\{1_A, 3_A, 5_A\}$. Define $f(2_B) = 5_A$.*
3) *Similarly, node 3_B has degree 4, and can only be mapped to the remaining nodes that have degree 2, $\{2_A, 4_A, 6_A\}$. However, since we have mapped node 1_B to 2_A, we can only map node 3_B to 4_A or 6_A. Let's select 6_A, i.e., $g(3_B) = 6_A$.*
4) *Finally, we define $g(4_B) = 4_A, g(5_B) = 1_A, g(6_B) = 3_A$.*

The above process can be demonstrated through the following figure:

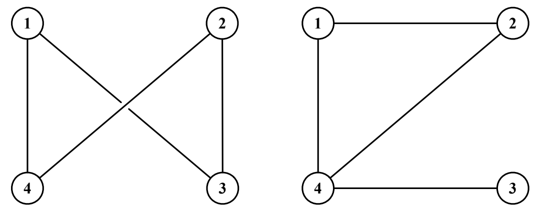


Corollary 4. *The corresponding rows and columns of the adjacency matrices of two isomorphic graphs have the same distribution of 0's and 1's.*

Example 6. *Let*

$$A = \begin{bmatrix} 1&0&1&1\\ 0&1&1&1\\ 1&1&1&0\\ 1&1&0&1 \end{bmatrix}, \quad B = \begin{bmatrix} 1&1&0&1\\ 1&1&0&1\\ 0&0&1&1\\ 1&1&1&1 \end{bmatrix}.$$

The corresponding graphs are:



The characteristic polynomials for matrix A is $x^4 - 4x^3 + 2x^2 + 4x - 3$ and the characteristic polynomials for matrix B is $x^4 - 4x^3 + 2x^2 + 2x$, which is different from that of A. Therefore, the two graphs are not isomorphic. The inequivalence of these two graphs can also be confirmed because the weights of the rows and columns in their adjacency matrices are different.

Theorem 3. *Let A and B be $n \times n$ adjacency matrices of graphs G_1 and G_2, respectively. Then G_1 and G_2 are isomorphic if and only if their adjacency matrices have the same characteristic polynomial (the same eigenvalues).*

The proof of this theorem will be provided in the full paper.

Corollary 5. *Let A and B be $n \times n$ adjacency matrices of graphs G_1 and G_2 that have the same set of eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$, all with single multiplicity. Let u_i and v_i be the normalized (length equal to 1) eigenvector corresponding to eigenvalue λ_i with respect to matrix A and B, respectively. Let $U = [u_1 \; \cdots \; u_n]$, $V = [v_1 \; \cdots \; v_n]$, $i = 1, \cdots, n$, and $P = V^T U$, then P is a permutation matrix such that*

$$PAP^T = B.$$

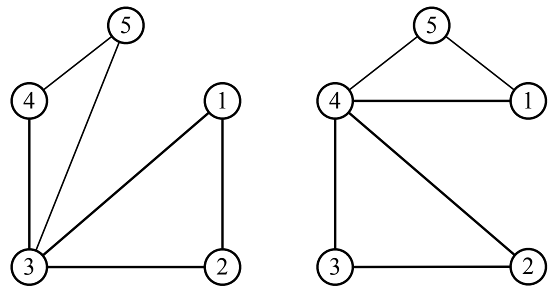
Corollary 5 provides an efficient algorithm to find a permutation matrix P such that

$$PAP^T = B,$$

when all eigenvalues are distinct, that is have single multiplicity. In case that the multiplicity of some eigenvalues are not single, even though the existence of such a permutation is known, the matrix derived in this way may not be a permutation matrix anymore.

However, when the multiplicities of some eigenvectors are not single, the result may not always be true, as shown in the following example.

Example 7. *For the following two graphs*



their corresponding adjacency matrices are given below:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The eigenvalues of these two matrices are $-1, -1, 1, 1, 4, 2$*, where both -1 and 1 are eigenvalues of multiplicity 2. Based on this order, we derive the orthonormal matrices from the eigenvectors of matrices* A *and* B *as follows:*

$$U = \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \\ \frac{\sqrt{2}}{2} & 0 & -\frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \\ 0 & 0 & 0 & -\frac{\sqrt{34+2\sqrt{17}}\sqrt{17}}{34} & \frac{\sqrt{34-2\sqrt{17}}\sqrt{17}}{34} \\ 0 & -\frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \\ 0 & \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \end{bmatrix},$$

and

$$V = \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \\ 0 & -\frac{\sqrt{2}}{2} & -\frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \\ 0 & \frac{\sqrt{2}}{2} & -\frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \\ 0 & 0 & 0 & -\frac{\sqrt{34+2\sqrt{17}}\sqrt{17}}{34} & \frac{\sqrt{34-2\sqrt{17}}\sqrt{17}}{34} \\ \frac{\sqrt{2}}{2} & 0 & \frac{1}{2} & \frac{\sqrt{34+2\sqrt{17}}(17-\sqrt{17})}{272} & \frac{\sqrt{34-2\sqrt{17}}(17+\sqrt{17})}{272} \end{bmatrix}.$$

It can be verified that

$$VU^T = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

is not even an integer matrix, let alone a permutation matrix.

However, it is easy to verify that we can construct a permutation matrix P *from the* 5×5 *identity matrix through 3 consecutive row interchanging operations:* $(2, 4), (3, 5), (4, 5)$*, that is for*

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

we have

$$PAP^T = B.$$

Based on Theorem 3, we can derive the following corollary.

Corollary 6. *The complexity in determining whether two undirected graphs are isomorphic is at most* $\mathcal{O}(n^3)$.

IV. Conclusion

In this paper, we analyzed the isomorphic problem of undirected graphs and presented two major theorems to characterize it. Specifically, we proved that determining whether two undirected graphs are isomorphic has a complexity of at most $\mathcal{O}(n^3)$. Additionally, we also designed algorithms to convert between isomorphic graphs along with multiple examples.

This work was supported in part by the National Science Foundation under Grant CCF-1919154 and Grant ECCS-1923409.

References

[1] Graph (discrete mathematics). [Online]. Available: https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)
[2] Z. Blumenfeld. Graph machine learning: An overview – key concepts for getting started. [Online]. Available: https://towardsdatascience.com/graph-machine-learning-an-overview-c996e53fab90
[3] M. Grohe and P. Schweitzer, "The graph isomorphism problem," *Communications of the ACM*, vol. 63, no. 11, p. 128–134, November 2020. [Online]. Available: https://doi.org/10.1145/3372123
[4] B. D. McKay and A. Piperno, "Practical graph isomorphism, ii," *Journal of Symbolic Computation*, pp. 94–112, January 2014. [Online]. Available: https://doi.org/10.1016/j.jsc.2013.09.003
[5] Graph isomorphism problem. [Online]. Available: https://en.wikipedia.org/wiki/Graph_isomorphism_problem
[6] R. Karp, "Reducibilities among combinatorial problems," in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, New York, 1972, p. 85–103.
[7] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
[8] Principal component analysis part 1: The different formulations. [Online]. Available: https://towardsdatascience.com/principal-component-analysis-part-1-the-different-formulations-6508f63a5553
[9] How are eigenvalues used in real life?
[10] Applications of eigenvalues and eigenvectors. [Online]. Available: https://www.intmath.com/matrices-determinants/8-applications-eigenvalues-eigenvectors.php
[11] Adjacency matrix. [Online]. Available: https://mathworld.wolfram.com/AdjacencyMatrix.html
[12] D. Margalit and J. Rabinoff. Interactive linear algebra. [Online]. Available: https://textbooks.math.gatech.edu/ila/ila.pdf