Some further development on the eigensystem approach for graph isomorphism detection

P.R. He\textsuperscript{a}, W.J. Zhang\textsuperscript{a,\textdagger}, Q. Li\textsuperscript{b}

\textsuperscript{a}Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Canada, SK S7N 5A9  
\textsuperscript{b}Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

Abstract

Many science and engineering problems can be represented by a network, a generalization of which is a graph. Examples of the problems that can be represented by a graph include: cyclic sequential circuit, organic molecule structures, mechanical structures, etc. The most fundamental issue with these problems (e.g., designing a molecule structure) is the identification of structure, which further reduces to be the identification of graph. The problem of the identification of graph is called graph isomorphism. The graph isomorphism problem is an NP problem according to the computational complexity theory. Numerous methods and algorithms have been proposed to solve this problem. Elsewhere we presented an approach called the eigensystem approach. This approach is based on a combination of eigenvalue and eigenvector which are further associated with the adjacency matrix. The eigensystem approach has been shown to be very effective but requires that a graph must contain at least one distinct eigenvalue. The adjacency matrix is not shown sufficiently to meet this requirement. In this paper, we propose a new matrix called adjusted adjacency matrix that meets this requirement. We show that the eigensystem approach based on the adjusted adjacency matrix is not only effective but also more efficient than that based on the adjacency matrix.

\textsuperscript{\textdagger}Corresponding author. Tel.: +1 306 966 5478; fax: +1 306 966 5427.  
E-mail addresses: peh985@mail.usask.ca (P.R. He), wjz485@engr.usask.ca (W.J. Zhang), mqli@ntu.edu.sg (Q. Li).
1. Introduction

Many science and engineering problems can be represented by a network, a generalization of which is a graph. Examples of the problems that can be represented by a graph include: cyclic sequential circuit, organic molecule structures [1,2], mechanical structures (especially kinematic chains) [3–9], etc. Specifically, in the recent years, graph theory has been successfully applied in the area called bioinformatics and computational biology [10,11]. The most fundamental issue with these problems (e.g., designing a molecule structure) is the identification of structure, which further reduces to be the identification of graph. The problem of the identification of graph is called graph isomorphism (GI). The GI problem is an NP problem according to the computational complexity theory.

The identification of GI is notably an NP (nondeterministic polynomial time) problem in the complexity class. Mathematicians and computer scientists have been working on finding efficient algorithms for GI [12,13]. So far, no known polynomial-time algorithm for GI has been found for general graphs. Excellent progresses have been made towards polynomial-time algorithms for special graphs, such as trees, interval graphs, planar graphs, and permutation graphs [14,15]. In this paper, concern is with general graphs.

The studies regarding the use of the characteristic polynomial of the adjacency matrix (AM) to characterize a graph in terms of isomorphism have a long history in graph theory [16,17]. The applications of the characteristic polynomial to detect isomorphism in kinematic chains have been discussed in [7–9]. The chief merit of this method is its computational simplicity and efficiency. However, this method only provides a necessary condition for GI.

Another stream of studies on GI is to develop a canonical code (if any) for a graph directly based on AM. Since the AM of a graph is always symmetric and has zero diagonal elements, its upper right triangular part is sufficient to describe the graph. Each row in the upper right triangular matrix can be regarded as a binary code, and these binary codes can be combined into an integer number called optimal code. This integer number uniquely corresponds to the graph and the labeling of the vertices of the graph. There exists a unique maximum number defined as max code of the graph [18], and there also exists a unique minimum code defined as min code of the graph [19]. Therefore, one can identify GI by comparing these unique or canonical codes of two graphs. However, the method based on such codes requires an exhaustive search in many worst cases. As a modification of the max code, Tang et al. [20] introduced a degree code, suggesting: first re-label the AM such that the degrees of the vertices of a graph are in a descending order, and then search for the max code based on the re-labeled AM. Note that the degree of a vertex of a graph is defined as the number of edges incident to the vertex. This method can thus reduce the search space for a canonical code. Luo et al. [21] developed an incident degree code approach. The incident degree code of a vertex of a graph is a number combined by the weights of the edges incident to that vertex and the degree codes of the vertices adjacent to that vertex. Suppose vertex $i$ in a graph has a degree $m$ and is connected to its adjacent $m$ vertices with weights $w_{i1}, w_{i2}, \ldots, w_{im}$. Each adjacent vertex has its own degree
denoted by \( d_{i1}, d_{i2}, \ldots, d_{in} \). The incident degree code is defined as 
\[ d_{i1}, d_{i2}, \ldots, d_{in}, w_{i1}, w_{i2}, \ldots, w_{in} \]
by following the arrangement 
\[ d_{i1} > d_{i2} > \cdots > d_{in} \).
If some adjacent vertices have the same degree, the weights of the corresponding 
edges should be in a descending order. It is obvious that the method of incident 
degree code uses more features of the vertices of a graph than the method of degree 
code, and thus the incident degree code method is generally more efficient than the 
degree code method. Based on the observation of a generic feature with these code-
based methods, i.e., imposing the constraints on vertices based on the topological 
feature of a graph prior to any permutation, Zhang and Li [22] proposed a method 
called vertex feature degree (VFD). The VFD method extends the method of incident 
degree by adding the body type of vertices into the incident degree code. In general, 
therefore, this method can further reduce the searching space for the identification of 
graphs. It is noted, however, that all these modified code methods can present the 
same defect as the methods of optimal code when the graphs to be detected are 
relatively regular or uniform.

Partitioning of a graph is another method used for GI. McKay [23] proposed a 
canonical labeling algorithm for the GI problem. Suppose that \( V \) denotes the vertex 
set \( \{1, 2, \ldots, n\} \) of a graph. A partition \( \pi \) of the set \( V \) is a set of disjoint non-empty 
subsets of \( V \): \( V_1, V_2, \ldots, V_m \). An ordered partition of \( V \) is a sequence 
\( (V_1, V_2, \ldots, V_r) \) such that \( \{V_1, V_2, \ldots, V_r\} \) is a partition of \( V \). Each element (called 
cell) of a partition of \( V \) will be further split until a leaf partition which is a partition 
with only singleton sets. There is an automorphism if two distinct leaf partitions 
result in the same AMs after re-labeling vertices. If we construct a label for a graph 
by concatenating the rows of the AM to form an \( n \times n \) binary number, we can then 
compute a canonical label for the graph by looking at the label of every 
automorphism of the graph, and returning the smallest one. This algorithm is 
basically a depth first search of the space of partitions, with the added optimization 
to refine each partition before expanding its children. Kim and Kwak [24] proposed 
an algorithm of topological ordering of vertices for generating a unique edge list of a 
given graph, and used it for the isomorphism test. A specific starting vertex, which 
has the highest degree, is used as the root of topological ordering in this algorithm. 
According to the distances from a starting vertex, the vertices can be partitioned into 
level structure. Some rules are used to determine the order of vertices in each level. 
Renumbering the vertices for the level structure results in a new edge list after the 
level structure is ordered completely. This new edge list (or a representative of new 
edge lists) is unique for a given graph, and can be used as identifier of isomorphism 
problem. However, the challenge also exists with the partition method when graphs 
are relatively regular and uniform. This is because the partition method also uses the 
degree as the ordering condition.

Elsewhere, we proposed a method which is based on a combination of eigenvalue 
and eigenvector, as opposed to relevant methods in the literature which only make 
use of the eigenvalue. This method is called the eigensystem approach. The 
eigensystem approach has been shown to be very effective but requires that a graph 
must contain at least one distinct eigenvalue. The AM is not shown sufficiently to 
meet this requirement. In this paper, we propose a new matrix called adjusted
adjacency matrix (AAM) that meets this requirement. We show that the eigensystem approach based on the adjusted adjacency matrix is not only effective but also more efficient than that based on the AM.

We arrange the remainder of this paper as follows. Section 2 gives an introduction of the eigensystem method, in which the condition for the eigensystem method to work is discussed. The AAM is proposed in Section 3 to meet this condition. A comparison of the AM-based and the AAM-based eigensystem methods is given in Section 4. The computational complexity of the proposed method is analyzed in Section 5. A conclusion is given in Section 6.

2. The eigensystem approach

We proposed in [25] a new approach based on quadratic forms for GI of the graphs that are simple, connected, undirected, and labeled graphs. The basic idea of this approach is to represent two graphs using two quadratic forms in which the matrices of the two quadratic forms are exactly the AMs of two graphs, and then compare their quadratic surfaces to determine if these two graphs are isomorphic. As it is known, the principal axes of the quadratic surface form a mutually orthogonal set. Assume that there is an orthogonal matrix \( \Psi \). The principal axes of the quadratic surface are chosen as a new co-ordinate system \( x'_1, x'_2, \ldots, x'_n \) by the transformation

\[
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
\end{bmatrix} = \Psi 
\begin{bmatrix}
    x'_1 \\
    x'_2 \\
    \vdots \\
    x'_n
\end{bmatrix}.
\]  

Hence, the quadratic form \( Q \) with the new variables is

\[
Q = [x_1 \ x_2 \ \cdots \ x_n]A = [x'_1 \ x'_2 \ \cdots \ x'_n]\Psi^T A \Psi = \lambda_1 x'_1^2 + \lambda_2 x'_2^2 + \cdots + \lambda_n x'_n^2, \tag{2}
\]

where
In the quadratic form, $1/\sqrt{\lambda_1}, 1/\sqrt{\lambda_2}, \ldots, 1/\sqrt{\lambda_n}$ are called the lengths that are commonly referred to as the semiaxes of the quadratic surface. The orthogonal matrix $\Psi$ represents the directions of these semiaxes. As a result, determining whether two graphs are isomorphic or not is equivalent to determining whether or not their quadratic surfaces are isomorphic, i.e., having the same semiaxes of quadratic surfaces and the directions of these semiaxes. Eq. (3) can be written as

$$A = \Psi \Lambda \Psi^T. \quad (4)$$

Eq. (4) indicates that matrix $A$ is the eigenvalue matrix of matrix $A$ and matrix $\Psi$ is the normalized eigenvector matrix of matrix $A$. Detection of isomorphic quadratic surfaces (i.e., isomorphic graphs) can be characterized as the detection of eigenvalues and eigenvectors of graphs. In graph theory, we normally call the eigenvalues of a graph the spectrum of the graph, and thus the graphs with the same spectrum are called cospectral graphs. It is further noted that in our previous work, we developed the following results based on the AM [25].

**Result 1.** For two graphs with all distinct eigenvalues, they are isomorphic if and only if they have the same graph spectrum and the equivalent eigenvectors.

**Result 2.** For two graphs with some coincident eigenvalues, they are isomorphic if they have the same graph spectrum and the equivalent ‘unique’ eigenvectors. This further implies that two graphs are definitely non-isomorphic if their graph spectrums are different or their ‘unique’ eigenvectors are not equivalent. It is noted that a unique eigenvector refers to the eigenvector associated with a distinct eigenvalue.

The above two results imply a very important concept, that is, the equivalent vector or matrix. Suppose the unique eigenvector set $X$ of graph $A$ is $X = [x_1, x_2, \ldots, x_m]$ and the unique eigenvector set $Y$ of graph $B$ is $Y = [y_1, y_2, \ldots, y_m]$. $X$ and $Y$ are equivalent if there exists a row permutation matrix $P$ such that

$$x_i = Py_i \quad \text{for } i = 1, 2, \ldots, m. \quad (5)$$

In the following, we describe the procedure to prove whether or not there is a $P$ matrix for a particular problem.

Take a pair of eigenvectors $x_i$ in $X$ and $y_i$ in $Y$. If there is a one-to-one mapping $\phi_i$ such that the $j$th element $x_{ij}$ in $x_i$ (which corresponds to the $j$th vertex of graph $A$) equals the $k$th element $y_{ik}$ in $y_i$ (which corresponds to the $k$th vertex of graph $B$) for all $j = 1, 2, \ldots, n$, i.e., $k = \phi_i(j)$, we can obtain a set of such mappings below.
where $\Phi$ is an $m \times n$ matrix. If all the rows of $\Phi$ in Eq. (6) are the same, there must be a common mapping function among mapping $\varphi_i$ (for $i = 1, 2, \ldots, m$). It is further noted that we can write $\varphi_i$ into the form of a row permutation matrix, i.e., $P$ in Eq. (5). In this case the $P$ matrix can be obtained. If not all the rows of $\Phi$ in Eq. (6) are the same, a common mapping function does not exist. This implies that the $P$ matrix does not exist.

It is noted that the mapping function $\varphi_j(i)$ could be a many-to-many relationship (i.e., given $j$ vertex in graph $A$ more than one vertex in graph $B$ corresponds to $j$; more than one vertex in graph $A$ may correspond to one vertex in graph $B$). We call this kind of mapping function $\varphi_j(i)$ the group-to-group mapping function. Table 1 gives an example. Take $\varphi_1(j)$ as an example. We can find that the vertex group $(2, 6, 8, 12)$ in graph $A$ has a many-to-many relationship with the vertex group $(2, 6, 8, 10)$ in graph $B$. Table 1 also shows that the mapping function $\varphi_j(i)$ may be different with respect to $j$. The many-to-many mapping implies more options of correspondence between two graphs. In particular, if the size of a many-to-many group is 3 the number of possible mappings will be 6. Let us denote $S_i$ as all mapping options for mapping function $i$. A common mapping set or function $(S_c)$ is defined by $S_c = S_1 \cap S_2 \cap S_3 \cap \ldots \cap S_m$. Table 1 shows an example of common mapping function. We further note that the common mapping function

Table 1
The concept of group-to-group mapping function $\varphi_i$ and its common mapping

<table>
<thead>
<tr>
<th>$i = 1$</th>
<th>$j$</th>
<th>1, 7</th>
<th>2, 6, 8, 12</th>
<th>3, 5, 9, 11</th>
<th>4, 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_1(j)$</td>
<td>1, 11</td>
<td>2, 4, 8, 10</td>
<td>3, 5, 7, 9</td>
<td>6, 12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i = 2$</th>
<th>$j$</th>
<th>1, 7</th>
<th>2, 6</th>
<th>8, 12</th>
<th>3, 5, 9, 11</th>
<th>4, 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_2(j)$</td>
<td>1, 11</td>
<td>8, 10</td>
<td>2, 4</td>
<td>3, 5, 7, 9</td>
<td>12</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i = 3$</th>
<th>$j$</th>
<th>1</th>
<th>7</th>
<th>2, 6</th>
<th>8</th>
<th>12</th>
<th>3, 9</th>
<th>5, 11</th>
<th>4, 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_3(j)$</td>
<td>1, 11</td>
<td>8, 10</td>
<td>4</td>
<td>2</td>
<td>3, 9</td>
<td>5, 7</td>
<td>6, 12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i = 4$</th>
<th>$j$</th>
<th>1, 7</th>
<th>2</th>
<th>6</th>
<th>8, 12</th>
<th>3</th>
<th>9</th>
<th>5</th>
<th>11</th>
<th>4, 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_4(j)$</td>
<td>1, 11</td>
<td>8</td>
<td>10</td>
<td>2, 4</td>
<td>3</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>6, 12</td>
<td></td>
</tr>
</tbody>
</table>

Common mapping

<table>
<thead>
<tr>
<th>$j$</th>
<th>1</th>
<th>7</th>
<th>2</th>
<th>6</th>
<th>8</th>
<th>12</th>
<th>3</th>
<th>9</th>
<th>5</th>
<th>11</th>
<th>4, 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi(j)$</td>
<td>1, 11</td>
<td>8</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>12</td>
<td>6</td>
</tr>
</tbody>
</table>
for the example shown in Table 1 contains one vertex only in each subgroup. This means that the common mapping function contains one option only. In such a case, we actually have reached a one-to-one correspondence between two graphs; that is to say two concerned graphs are isomorphic. It is possible that a common mapping function has a many-to-many relationship. For example, if we remove mapping \( \varphi_2 \) (the 2nd row) from Table 1, then the common mapping is not a one-to-one relationship (see the last column in Table 1).

An algorithm which is able to search all the possibilities of subdivisions in attempt to find a one-to-one common mapping for all group-to-group mappings has been developed in [25]. When the algorithm ends without finding any one-to-one mapping, we can conclude that the \( P \) matrix does not exist and therefore two corresponding graphs are not isomorphic.

### 3. Adjusted adjacency matrix of graph

The eigensystem approach so far discussed requires that there exists at least one distinct eigenvector in the AM of a graph. Nevertheless, this condition is not always satisfied with AM as representation of a graph. The primary objective of the present paper is to provide a solution to this problem. Our general idea is to define a new matrix to represent a graph. This new matrix must have at least one distinct eigenvalue. The new matrix is defined by extending the AM of a graph, and is given as follows:

\[
d_{ij} = \begin{cases} 
\text{same as the adjacency matrix,} & i \neq j, \\
\deg(v_i) \neq 0, & i = j.
\end{cases}
\]

We call the new matrix the AAM. The difference between an AM and an AAM is that all elements on the principal diagonal of the AM are zero but nonzero of the AAM. It is easy to see that AAM is unique and sufficient to represent a graph. Therefore, the isomorphism of two graphs can be studied based on their AAMs. We have the following theorem for the AAM.

**Theorem.** The AAM of a connected graph with \( n \) vertices will have at least one distinct eigenvalue, i.e., \( n \) which is the total number of vertices in this graph.

**Proof.** Consider the matrix

\[
B = \frac{1}{n} M,
\]

where \( M \) is an AAM of a connected graph. To prove that \( n \) is a distinct eigenvalue of matrix \( M \), we only need to prove that matrix \( B \) has a unique eigenvalue 1. It is easy to see that matrix \( B \) has two properties:

1. \( b_{ij} \geq 0; \)
2. \( \sum_{i=1}^{n} b_{ij} = \sum_{j=1}^{n} b_{ij} = 1; \quad i = 1, 2, \ldots, n; \; j = 1, 2, \ldots, n. \)
Therefore, matrix $B$ is a doubly stochastic matrix, and it has a largest eigenvalue $1$ and an eigenvector $[1 \ 1 \ \cdots \ 1]^T$ corresponding to the largest eigenvalue [26]. Since the associated graph of matrix $M$ (also matrix $B$) is strongly connected (for each entry $(i,j)$ in matrix $M$ there exists an integer $k$), matrix $B$ is an irreducible matrix. More exactly, matrix $B$ is an irreducible nonnegative matrix according to property (1) of the matrix. It is known that an irreducible nonnegative matrix with a nonzero principal diagonal is primitive [27], and an $n \times n$ nonnegative primitive matrix has a unique eigenvalue, 1, which is the largest among all its eigenvalues [28]. According to the definition of matrix $B$, we can conclude that matrix $M$ has a distinct eigenvalue (which is the largest), and the eigenvector associated with that eigenvalue is $[1 \ 1 \ \cdots \ 1]^T$. □

An example shown in Fig. 1 may help to explain this theorem further. Fig. 1 shows a five-vertex graph and its AAM. The AAM of the graph has the graph spectrum $A = \mathrm{diag}(5.0000, 3.6180, 3.6180, 1.3820, 1.3820)$ in which 5.0000 is the only distinct eigenvalue and its associated eigenvector has the same element 1 (or 0.4472 as a normalized value).

The advantage of the AAM of a graph is also that the GI detection based on AAM is more efficient than that based on AM in the sense that two nonisomorphic cospectral graphs based on AM may not be the cospectral graph based on AAM and the converse is not true. This is, however, just a conjecture. More work needs to be done to prove this conjecture. Examples supporting the conjecture will be discussed in the next section.

4. Comparison of AAM- and AM-based methods

4.1. Graphs not cospectral on AM and AAM

Fig. 2 includes two graphs, and their spectrums based on AM are

$$A_a = \mathrm{diag}(3, 2, 1.4142, 1.4142, 1, 0, 0, -1, -1.4142, -1.4142, -2, -3),$$

$$A_b = \mathrm{diag}(3, 2, 2, 1, 0, 0, 0, -1, -2, -2, -3).$$

Since the spectrums between the two graphs are different these two graphs are not cospectral. Therefore, these two graphs are not isomorphic. The spectrums of the
two graphs based on AAM are shown below:

\[ \Lambda_a' = \text{diag}(12, 11, 10.4142, 10.4142, 10, 9, 9, 8, 7.5858, 7.5858, 7, 6), \]
\[ \Lambda_b' = \text{diag}(12, 11, 11, 10, 9, 9, 9, 8, 7, 7, 6). \]

It is clear that these two graphs are not cospectral based on AAM also.

4.2. Graphs cospectral on AM but not on AAM

There are two cospectral kinematic chains on AM shown in Fig. 3, and their spectrums based on AM are as follows:

\[ \Lambda_a = \Lambda_b = \text{diag}(2.7523, 1.6037, 1.1354, 1.0000, 0.4528, 0.0000, \]
\[ -0.9182, -1.5279, -2.0693, -2.4289). \]

However, their graph spectrums based on AAM are
\[ A'_a = \text{diag}(10.0000, 9.2295, 8.7930, 8.5509, 7.6967, 7.5416, 6.7547, 5.9379, 5.1341, 4.3617) \]
\[ A'_b = \text{diag}(10.0000, 9.3134, 8.7685, 8.2729, 8.0000, 7.4351, 6.8406, 5.8922, 5.0782, 4.3990) \]

respectively, and they are clearly not cospectral.

Fig. 4 shows other two pairs of kinematic chains with 10 bars that are not isomorphic. Each pair of the kinematic chains is cospectral based on AM but is not cospectral based on AAM. Fig. 5 shows that two graphs are two cospectral trees with eight vertices. They are absolutely nonisomorphic trees but have the same graph spectrum based on AM:

\[ A_a = A_b = \text{diag}(2.3028, 1.3028, 0.0000, 0.0000, 0.0000, 0.0000, -1.3028, -2.3028) \]

However, they have different spectrums based on AAM, i.e.,

\[ A'_a = \text{diag}(8.0000, 7.7226, 7.0000, 7.0000, 7.0000, 6.4932, 4.8390, 1.9452) \]
\[ A'_b = \text{diag}(8.0000, 7.6458, 7.0000, 7.0000, 7.0000, 7.0000, 4.0000, 2.3542) \]

Two tree graphs are thus not isomorphic. Fig. 6 shows another two examples

Fig. 4. Two pairs of cospectral kinematic chains based on AM.

Fig. 5. Two cospectral trees with eight vertices based on AM.
where the graphs, respectively, in Fig. 6(a) and (b) are cospectral based on AM but not cospectral based on AAM and therefore they are not isomorphic. It is worth to mention that for all the graphs listed in [16,17], we have checked that they are not cospectral based on AAM. Intuitively, therefore, using AAM seems more efficient than using AM for detection of GI.

4.3. Graphs cospectral on both AM and AAM

We have observed that not all nonisomorphic graphs (which are cospectral based on AM) are not cospectral based on AAM. Fig. 7 shows two non-isomorphic kinematic chains with 12 bars. The spectrums of the two graphs based on AM are

\[
A_a = A_b = \text{diag}(2.7580, 1.7668, 1.5713, 1.4142, 0.6350, 0.4150, 0, -0.9815, -1.4142, -1.5382, -2, -2.6264)
\]

and the spectrums based on AAM are

\[
A'_a = A'_b = \text{diag}(12, 11.2652, 10.8953, 10.8019, 10.1955, 9.7365, 9.4450, 8.4181, 7.7701, 7.7530, 7.2414, 6.4779).
\]
Further examination of these two graphs shows their \( \psi \) matrices below, respectively,

\[
\begin{bmatrix}
0.2887 & -0.3236 & -0.3028 & -0.5211 & 0.0967 & 0.2312 & 0.4179 & -0.1984 & -0.2544 & -0.2319 & -0.1831 & 0.1408 \\
0.2887 & -0.3236 & -0.3028 & 0.5211 & 0.0967 & 0.2312 & -0.4179 & -0.1984 & -0.2544 & 0.2319 & -0.1831 & 0.1408 \\
0.2887 & -0.3623 & 0.4756 & 0.0000 & -0.4582 & -0.3010 & -0.0000 & 0.3017 & -0.3751 & -0.0000 & 0.0071 & 0.1639 \\
0.2887 & 0.4688 & -0.1858 & -0.0000 & -0.6210 & 0.1652 & 0.0000 & -0.3446 & -0.0975 & 0.0000 & 0.3467 & 0.0302 \\
0.2887 & -0.4457 & -0.0685 & 0.0000 & -0.2216 & 0.2193 & -0.0000 & 0.1636 & 0.7187 & 0.0000 & 0.2042 & -0.1766 \\
0.2887 & 0.0362 & -0.2025 & -0.4179 & 0.2405 & -0.2802 & -0.2319 & 0.1504 & -0.1514 & 0.5211 & 0.3010 & -0.3192 \\
0.2887 & 0.2390 & -0.3118 & -0.0000 & -0.1172 & -0.5365 & -0.0000 & 0.0753 & 0.3254 & 0.0000 & -0.5395 & 0.2412 \\
0.2887 & 0.1667 & 0.2307 & -0.2319 & 0.3080 & 0.0990 & -0.5211 & 0.0356 & 0.1152 & -0.4179 & 0.1933 & 0.4232 \\
0.2887 & 0.0362 & -0.2025 & 0.4179 & 0.2405 & -0.2802 & 0.2319 & 0.1504 & -0.1514 & -0.5211 & 0.3010 & -0.3192 \\
0.2887 & -0.0127 & 0.4944 & 0.0000 & 0.1320 & -0.1399 & 0.0000 & -0.6408 & 0.1177 & 0.0000 & -0.2239 & -0.4006 \\
0.2887 & 0.3542 & 0.1454 & 0.0000 & -0.0042 & 0.4930 & 0.0000 & 0.4697 & -0.1080 & -0.0000 & -0.4170 & -0.3476 \\
0.2887 & 0.1667 & 0.2307 & 0.2319 & 0.3080 & 0.0990 & 0.5211 & 0.0356 & 0.1152 & 0.4179 & 0.1933 & 0.4232
\end{bmatrix}
\]
$$\psi_b = \begin{bmatrix}
-0.2887 & 0.3236 & -0.3028 & -0.5211 & 0.0967 & 0.2312 & 0.4179 & -0.1984 & -0.2544 & -0.2319 & 0.1831 & 0.1408 \\
-0.2887 & 0.3236 & -0.3028 & 0.5211 & 0.0967 & 0.2312 & -0.4179 & -0.1984 & -0.2544 & 0.2319 & 0.1831 & 0.1408 \\
-0.2887 & -0.4688 & -0.1858 & 0.0000 & -0.6210 & 0.1652 & 0.0000 & -0.3446 & -0.0975 & -0.0000 & -0.3467 & 0.0302 \\
-0.2887 & 0.3623 & 0.4756 & -0.0000 & -0.4582 & -0.3010 & -0.0000 & 0.3017 & -0.3751 & -0.0000 & -0.0071 & 0.1639 \\
-0.2887 & 0.1802 & -0.4840 & 0.0000 & 0.0637 & -0.1006 & -0.0000 & 0.5526 & 0.1491 & 0.0000 & -0.5150 & -0.2073 \\
-0.2887 & -0.2390 & -0.3118 & 0.0000 & -0.1172 & -0.5365 & -0.0000 & 0.0753 & 0.3254 & 0.0000 & 0.5395 & 0.2412 \\
-0.2887 & -0.2527 & 0.0789 & 0.0000 & 0.4172 & -0.4598 & -0.0000 & -0.2518 & -0.4519 & -0.0000 & -0.0869 & -0.4312 \\
-0.2887 & -0.1667 & 0.2307 & -0.2319 & 0.3080 & 0.0990 & -0.5211 & 0.0356 & 0.1152 & -0.4179 & -0.1933 & 0.4232 \\
-0.2887 & -0.1667 & 0.2307 & 0.2319 & 0.3080 & 0.0990 & 0.5211 & 0.0356 & 0.1152 & 0.4179 & -0.1933 & 0.4232 \\
-0.2887 & -0.3542 & 0.1454 & -0.0000 & -0.0042 & 0.4930 & 0.0000 & 0.4697 & -0.1080 & -0.0000 & 0.4170 & -0.3476 \\
-0.2887 & 0.2292 & 0.2129 & -0.4179 & -0.0448 & 0.0397 & -0.2319 & -0.2386 & 0.4182 & 0.5211 & 0.0098 & -0.2886 \\
-0.2887 & 0.2292 & 0.2129 & 0.4179 & -0.0448 & 0.0397 & 0.2319 & -0.2386 & 0.4182 & -0.5211 & 0.0098 & -0.2886
\end{bmatrix}$$
We can apply our proposed method outlined in Section 2 to this problem, and can conclude that there is no common mapping between two eigenvector sets. Thus, there is no permutation matrix \( P \); in other words, the AAMs of two graphs are not equivalent. Therefore, these two kinematic chains are not isomorphic.

Fig. 8 shows two graphs with fifteen vertices. They are cospectral both based on AM and AAM. Applying the procedure described in Section 2, we can show that there is no matrix \( P \) either based on AM or on AAM. Therefore, we can conclude that the two graphs are not isomorphic.

Fig. 9 shows two graphs that are cospectral based on AM and are isomorphic. We can apply our proposed procedure to this problem and can conclude that they are isomorphic based on AAM also.

5. Computational complexity of the approach

We now analyze the computational complexity of the eigensystem approach for GI. The time complexity for a single detecting loop with the proposed algorithm is dominated by eigendecomposition of AAMs, comparisons of eigenvalues and the eigenvectors corresponding to the distinct eigenvalues between graphs. The time cost
for the comparisons of eigenvalues–eigenvectors of two graphs is mainly for sorting eigenvalues and sorting each eigenvector corresponding to a distinct eigenvalue. Once the common mapping has a many-to-many relationship (i.e., a group-to-group mapping), a recursive procedure of the eigensystem approach has to be performed, which consists of several similar detecting loops.

As it is known, in general, the computational complexity of eigendecomposition with a $N \times N$ matrix is $O(N^3)$, while the computational complexity of sorting problem for $N$ elements is $O(N \log N)$. The time complexity of comparisons of eigenvalues–eigenvectors is thus

$$T = \text{sorting eigenvalues} + \text{number of eigenvectors compared}$$

where the number of eigenvectors compared is $N$ for $N$ vertex graph in the worst case. The computational complexity of comparisons of eigenvalues–eigenvectors, therefore, is bounded by $O(N^2 \log N)$. Finally, the computational complexity of a single detecting loop by the eigensystem approach can be bounded by $O(N^3)$.

Suppose that a determination of whether two graphs are isomorphic or not could be made after $M$ detecting loops, then the complexity of the eigensystem approach for GI is $O(MN^3)$.

6. Conclusion

GI is a common problem in many science and engineering applications. The study on efficient algorithms for GI has a long history, and many methods have been proposed in the literature. In [25], we proposed a new method for GI. This new method is based on a combination of eigenvalues and eigenvectors which are further associated with the adjacency matrix (AM) of a graph. The new method is called the eigensystem approach. In [29] we further improved this method yet left only one requirement on this method; that is, the AM of a graph must contain at least one distinct eigenvalue.

We observed that the AM cannot ensure this requirement and therefore we define a new matrix representation for a graph, which may be closely related to the AM but meets that requirement. This paper presented such a matrix called adjusted adjacency matrix (AAM). We proved that the AAM can meet that requirement and showed that the eigensystem approach based on the AM [25,29] can be readily migrated to the one based on the AAM. We conducted the computational complexity for the eigensystem approach based on the AAM. There are the following conclusions that can be derived from the paper:

1. The AAM of a graph contains at least one distinct eigenvalue.
2. The algorithm based on the AAM is more efficient than that based on the AM because of a proposed conjecture.
3. We have so far not found any counterexample for the eigensystem approach.
(4) The computational complexity of the eigensystem approach based on the AAM
is bounded by $O(MN^3)$, where $N$ is the number of vertices in a graph, and $M$ is
the level of a recursive algorithm in the method can go in searching a common
mapping function.

We further suggest a conjecture based on our experience in many case studies: two
cospectral graphs based on the AM may not be cospectral based on the AAM;
whereas the converse is not true.

Acknowledgements

The research presented in this paper has been partially supported by Natural
Sciences and Engineering Research Council of Canada (NSERC) through an
NSERC operating grant awarded to the second author.

References

[3] L. Dobrjanskyi, F. Freudenstein, Some applications of graph theory to the structural analysis of
[5] W.J. Sohn, F. Freudenstein, An application of dual graphs to the automatic generation of the
[8] H.S. Yan, A.S. Hall, Linkage characteristic polynomials: definitions, coefficients by inspection,
[9] T.S. Mruthyunjaya, H.R. Balasubramanian, In quest of a reliable and efficient computational test for
[10] M. Kanehisa, Graph comparison and path computation methods for predicting molecular networks
from genome information, in: Keynote on the 8th International Conference on Intelligent System for
Molecular Biology ISMB 2000, San Diego, California, USA, August 2000.
application to detect functionally related enzyme clusters, Nucleic Acids Res. 28 (20) (2000)
4021–4028.
[14] H.L. Bodlaender, Polynomial algorithms for graph isomorphism and chromatic index on partial
63–77.