Journal of Mathematical Chemistry 8(1991)19–37

ON GRAPH ISOMORPHISM AND GRAPH AUTOMORPHISM*

Tomislav P. ŽIVKOVIĆ
Rudjer Bošković Institute, P.O. Box 1016, 41001 Zagreb, Croatia, Yugoslavia

Abstract

The problem of graph isomorphism, graph automorphism and a unique graph ID is considered. A new approach to the solution of these problems is suggested. The method is based on the spectral decomposition $\mathbf{A} = \sum_i \lambda_i \mathbf{K}_i$ of the adjacency matrix A. This decomposition is independent of the particular labeling of graph vertices, and using this decomposition one can formulate an algorithm to derive a canonical labeling of the corresponding graph G. Since the spectral decomposition uniquely determines the adjacency matrix A and hence graph G, the obtained canonical labeling can be used in order to derive a unique graph ID. In addition, if the algorithm produces several canonical labelings, all these labelings and only these labelings are connected by the elements of the graph automorphism group $\mathcal{G}$. In this way, one obtains all elements of this group. Concerning graph isomorphism, one can use a unique graph ID obtained in the above way. However, the algorithm to decide whether graphs G and G' are isomorphic can be substantially improved if this algorithm is based on the direct comparison between spectral decompositions of the corresponding adjacency matrices A and A'.

1. Introduction

The problems of graph isomorphism, graph automorphism and a unique graph ID are probably the three most important graph-theoretical problems [1] and they have been extensively studied in the literature [1–24]. The solution of these problems is important for chemical documentation and nomenclature, for the identification of equivalent atoms in a molecule, for the enumeration of different isomers, etc. These problems are also important in other areas of science, such as statistical mechanics and the theory of disordered structures. It is generally quite difficult to assert that two graphs have an identical connectivity, to find the graph isomorphism group, or to formulate an efficient method for the derivation of a unique graph ID [1–24]. In this paper, we suggest a new method for the solution of these problems.

Each graph G can be represented by the adjacency matrix A. This representation depends on the particular labeling of graph vertices. If G is an n-order graph and if A is the corresponding adjacency matrix, then another adjacency matrix can be obtained if graph vertices (s) are relabeled in some other order [1]. There are $n!$ possible labelings of vertices (s), and hence there are generally $n!$ adjacency matrices

*Research supported by the Yugoslav Ministry for Development (Grant P-339).

A associated with the same graph G. This number decreases if G has a nontrivial automorphism group $\mathcal{G}$.

The graph isomorphism problem can be stated in the following way: given adjacency matrices **A** and **A'**, find an efficient algorithm to decide whether or not these matrices represent the same graph G. If matrices **A** and **A'** represent the same graph, they differ only in some permutation of graph vertices. Hence, matrices **A** and **A'** represent the same graph if and only if there exists a permutation matrix **P** such that $\mathbf{A} = \mathbf{P}\mathbf{A}'\mathbf{P}^{+}$. Since the permutation **P** can be considered to permute graph vertices, one finds that adjacency matrices **A** and **A'** represent the same graph if and only if there exists a permutation **P** such that

$$\mathbf{A}_{sp} = \mathbf{A}'_{\mathbf{P}s,\mathbf{P}p} \qquad (s, p = 1, \ldots, n), \tag{1}$$

where **P** now permutes indices s and p.

Graph automorphism is the isomorphism of a graph with itself [1]. If the graph G has a nontrivial automorphism group $\mathcal{G}$, then each permutation $\mathbf{P} \in \mathcal{G}$ leaves the adjacency matrix **A** invariant. Hence,

$$\mathbf{A}_{sp} = \mathbf{A}_{\mathbf{P}s,\mathbf{P}p} \qquad (s, p = 1, \ldots, n) \tag{1'}$$

for each permutation $\mathbf{P} \in \mathcal{G}$. Inversely, if a permutation **P** satisfies (1'), then $\mathbf{P} \in \mathcal{G}$ is an element of the automorphism group $\mathcal{G}$.

An early systematic attempt to solve the graph isomorphism problem is the node-to-node search [2]. This approach requires extensive bookkeeping of the examined possibilities, and it is impractical already for graphs of medium size. Alternative methods utilize graph spectrum [3], distribution of valences of vertices, examination of edge types, comparison of characteristic polynomials of graph subgraphs [4,5], etc. Particularly interesting are various approaches based on the spectral decomposition of the adjacency matrix [6–8]. Thus, Cvetković considers angles between the eigenspaces of the adjacency matrix and the axes of the corresponding real vector space [7,8]. He uses these invariances in conjuncture with the graph spectrum in order to improve the discrimination between nonisomorphic graphs.

The problem of the unique graph identification is usually treated as a search for a "canonical" labeling of graph vertices [9–14]. Particularly simple is the approach suggested by Morgenau and Murphy [10]. Their method involves a repeated multiplication of a column vector $\boldsymbol{U}_k$ by the adjacency matrix **A** to generate a new column vector $\boldsymbol{U}_{k+1}$. The initial vector $\boldsymbol{U}_1$ is chosen to have all elements equal to unity. Graph vertices are then divided into equivalence classes according to the coefficients of the vector $\boldsymbol{U}_k$ as $k \to \infty$. This produces a canonical labeling of a graph G [10]. Normalized vectors $\boldsymbol{U}_k$ are known to converge to the eigenvector Ψ_1 of the adjacency matrix **A** which corresponds to the largest eigenvalue λ_1 of **A**. The resulting canonical labeling hence depends on the coefficients of the single eigenvector Ψ_1. There is a similar approach based on the concept of the extended

connectivity [11, 12]. This approach produces exactly the same result [11–14]. Moreover, the extended connectivity numbers are the same as those obtained by the enumeration of walks of corresponding length, generated by powers of **A** [11–14].

Due to its simplicity, the above approach is quite appealing, but it is not reliable since it does not guarantee an efficient solution of the graph ID problem. A single eigenvector Ψ_1 is not sufficient for the unique determination of the adjacency matrix, and hence the resulting canonical labeling is not guaranteed to be complete. In order to guarantee a successful termination of the algorithm, one has to include in a graph invariant way all eigenvalues of the corresponding adjacency matrix. The construction of such an algorithm which would guarantee a successful termination and which would not require an unnecessarily large operation count is a major problem.

2. The method

The main idea of the method suggested here can be illustrated in the following way. Assume that λ is a nondegenerate eigenvalue of the adjacency matrix **A** of a graph G, and let $\Psi = \sum_s c_s \Phi_s$ be a normalized eigenvector of **A**. Here, $c_s \equiv \langle s \mid \Psi \rangle$ is a coefficient of Ψ at vertex (s) of G, and $\Phi_s \equiv |s\rangle$ is the corresponding vector. Since **A** is real symmetric, all coefficients c_s can be chosen to be real, and with this choice the normalized eigenvector Ψ is fixed up to the sign ± 1. One can usually determine this sign by forming a descending sequence $\{c_s\}$ of the coefficients c_s and a descending sequence $\{-c_s\}$ of the coefficients $-c_s$, and by choosing the sequence which is lexicographically larger. For a given λ, the sequence determined in this way is independent of the initial graph labeling, and hence this sequence can be used in order to obtain a canonical labeling of G. The labeling obtained in this way is ambiguous for these vertices, which have the same coefficients c_s, and we consider all labelings which can be derived by permuting vertices with the same coefficients to be admissable canonical labelings of G. If the sequences $\{c_s\}$ and $\{-c_s\}$ are lexicographically equivalent, both sequences are used in order to obtain admissable labelings of G. Since it is very unlikely for nonequivalent vertices of G to have identical coefficients c_s, such an approach can drastically reduce the number of admissable labelings.

A single eigenvector Ψ does not determine the adjacency matrix **A**, and hence the above simple approach does not guarantee either the construction of a unique graph ID, or the solution of the graph isomorphism and graph automorphism problem. In order to eliminate this deficiency, we generalize in a systematic way the above idea to all eigenvectors of **A**, in particular to all degenerate eigenvectors [25].

The adjacency matrix **A** can be written in the form

$$\mathbf{A} = \sum_{i,\nu} \lambda_i |\Psi_{i\nu}\rangle\langle\Psi_{i\nu}| = \sum_i \lambda_i \boldsymbol{K}_i , \tag{2a}$$

where λ_i are eigenvalues of **A**, while $|\Psi_{iv}\rangle$ are the corresponding orthonormalized eigenvectors. The index v is a degeneracy index, and it labels different eigenvectors $|\Psi_{iv}\rangle$ corresponding to the same eigenvalue λ_i. Relation (2a) is a spectral decomposition [26] of a matrix **A**, and the sum

$$\boldsymbol{K}_i = \sum_v |\Psi_{iv}\rangle\langle\Psi_{iv}| \tag{3a}$$

is a projection operator which projects on a subspace $\boldsymbol{X}_i$ spanned by the degenerate eigenvectors $|\Psi_{iv}\rangle$ ($v = 1, 2, \ldots$). According to (2a), the matrix elements $\mathbf{A}_{sp} \equiv \langle s \,|\, \mathbf{A} \,|\, p\rangle$ of the adjacency matrix **A** are given by

$$\mathbf{A}_{sp} = \sum_{i,v} \lambda_i \langle s|\Psi_{iv}\rangle\langle\Psi_{iv}|p\rangle \equiv \sum_i \lambda_i \langle s|\boldsymbol{K}_i|p\rangle, \tag{2b}$$

where $\langle s \,|\, \Psi_{iv}\rangle$ is a component of a normalized eigenvector $|\Psi_{iv}\rangle$ on a vertex (s). Similarly, the matrix elements $\mathbf{A}'_{sp} \equiv \langle s \,|\, \mathbf{A}' \,|\, p\rangle$ of another adjacency matrix $\mathbf{A}'$ can be written in the form

$$\mathbf{A}'_{sp} = \sum_{i,v} \lambda'_i \langle s|\Psi'_{iv}\rangle\langle\Psi'_{iv}|p\rangle \equiv \sum_i \lambda'_i \langle s|\boldsymbol{K}'_i|p\rangle, \tag{2c}$$

where the apostrophe (') refers to the quantities related to the matrix $\mathbf{A}'$. If **A** and $\mathbf{A}'$ represent the same graph G, then according to (1) there exists a permutation **P** such that

$$\sum_i \lambda_i \langle s|\boldsymbol{K}_i|p\rangle = \sum_i \lambda'_i \langle \mathbf{P}s|\boldsymbol{K}'_i|\mathbf{P}p\rangle, \qquad (s, p = 1, \ldots, n). \tag{4}$$

Since eigenvectors $|\Psi_{iv}\rangle$ of **A** as well as eigenvectors $|\Psi'_{iv}\rangle$ of $\mathbf{A}'$ are linearly independent, relation (4) is satisfied if and only if $\lambda_i = \lambda'_i$ ($i = 1, \ldots$) and if in addition $\boldsymbol{K}_i = \mathbf{P}\boldsymbol{K}'_i\mathbf{P}^+$ ($i = 1, 2, \ldots$). Hence:

THEOREM 1

Let $\mathbf{A} = \sum_i \lambda_i \boldsymbol{K}_i$ and $\mathbf{A}' = \sum_i \lambda'_i \boldsymbol{K}'_i$ be spectral decompositions of adjacency matrices **A** and $\mathbf{A}'$, respectively. Then **A** and $\mathbf{A}'$ represent the same graph G if and only if $\lambda_i = \lambda'_i$ ($i = 1, \ldots, n$) and if in addition there exists a permutation matrix **P** such that

$$\boldsymbol{K}_i = \mathbf{P}\boldsymbol{K}'_i\mathbf{P}^+ \qquad (i = 1, 2, \ldots). \tag{5}$$

This second condition can be written explicitly in terms of matrix elements of projection operators $\boldsymbol{K}_i$ and $\boldsymbol{K}'_i$

$$\langle s \mid \boldsymbol{K}_i \mid p \rangle = \langle \mathbf{P}s \mid \boldsymbol{K}'_i \mid \mathbf{P}p \rangle, \qquad (s, p = 1, \ldots, n), \tag{5'}$$

where $\mathbf{P}$ now permutes vertices (s) and (p).

Theorem 1 addresses the problem of graph isomorphism. In a similar way, one derives from the relation (1′) the following theorem which refers to graph automorphism:

THEOREM 2

Let $\mathbf{A}$ be the adjacency matrix of a graph G, and let $\mathbf{A} = \sum_i \lambda_i \boldsymbol{K}_i$ be a spectral decomposition of $\mathbf{A}$. Further, let $\mathcal{G}$ be the automorphism group of G and let $\mathbf{P} \in \mathcal{G}$ be an element of this group. Then each projection operator $\boldsymbol{K}_i$ satisfies

$$\langle s \mid \boldsymbol{K}_i \mid p \rangle = \langle \mathbf{P}s \mid \boldsymbol{K}_i \mid \mathbf{P}p \rangle, \qquad (s, p = 1, \ldots, n). \tag{6}$$

Inversely, if the permutation $\mathbf{P}$ satisfies (6) for each projection operator $\boldsymbol{K}_i$, then $\mathbf{P} \in \mathcal{G}$ is an element of the automorphism group $\mathcal{G}$.

The first condition $\lambda_i = \lambda'_i$ in theorem 1 is the requirement that the adjacency matrices $\mathbf{A}$ and $\mathbf{A}'$ should be isospectral. This requirement is not sufficient for graph isomorphism, as well documented in the literature [27]. The requirement of isospectrality should be supplemented with the requirement (5), and only then are matrices $\mathbf{A}$ and $\mathbf{A}'$ guaranteed to represent the same graph G.

Condition (5) can be used to generalize the basic idea stated in the beginning of this section to all eigenvectors of $\mathbf{A}$ and $\mathbf{A}'$. Formally, this condition replaces the single condition (1) with a set of conditions (5). This allows for a construction of much more efficient algorithms. Using different pairs $\{\boldsymbol{K}_i, \boldsymbol{K}'_i\}$ of projection operators, one can perform many different tests on isomorphism. If any of these tests fail, the algorithm terminates and the corresponding adjacency matrices do not represent isomorphic graphs. In addition, each of these tests is very selective, since matrix elements of projection operators usually discriminate between nonequivalent vertices. In this way, one can systematically reduce the number of admissable graph labelings. Projection operators $\boldsymbol{K}_i$ are hence much more powerful in discriminating between nonisomorphic graphs than the adjacency matrix $\mathbf{A}$, whose matrix elements can assume only two values, zero and one. More precisely, using projection operators instead of the adjacency matrix, one can easily eliminate a huge number of permutations $\mathbf{P}$ which are not admissable.

Before considering various algorithms for graph isomorphism, graph automorphism and a unique graph ID in more detail, let us briefly mention that from theorems 1 and 1′ one can deduce some useful *necessary* conditions for graph isomorphism and graph automorphism.

Consider diagonal elements $\langle s \mid \boldsymbol{K}_i \mid s \rangle$ $(s = 1, \ldots, n)$ of the projection operator $\boldsymbol{K}_i$. There exists a permutation $\mathbf{P}$ which permutes vertices (s) in such a way that

matrix elements $\langle \mathbf{P}s \mid \boldsymbol{K}_i \mid \mathbf{P}s \rangle$ ($s = 1, \ldots, n$) form a descending sequence. Let $\mathcal{D}(\boldsymbol{K}_i)$ denote this sequence. In other words, $\mathcal{D}(\boldsymbol{K}_i)$ is an ordered set which contains diagonal elements of $\boldsymbol{K}_i$ in descending order. This set does not depend on the initial labeling of graph vertices, i.e. it is a graph invariant. From theorem 1 now follows:

COROLLARY 1

A necessary condition for graphs G and G' to be isomorphic is that

$$\mathcal{D}(\boldsymbol{K}_i) = \mathcal{D}(\boldsymbol{K}_i') \qquad (i = 1, 2, \ldots). \tag{7}$$

Diagonal elements of projection operators are related to graph angles considered by Cvetković in his analysis of graph isomorphism [7,8]. He has also shown that a graph spectrum and graph angles do not form a complete set of graph invariants [7,8]. Klein has independently utilized relation (7) in order to determine graph isomorphism of fullerenes [28].

The sets $\mathcal{D}(\boldsymbol{K}_i)$ can be used to facilitate the construction of a canonical labeling of G. In particular, if two vertices s and p are equivalent, then

$$\langle s \mid \boldsymbol{K}_i \mid s \rangle = \langle p \mid \boldsymbol{K}_i \mid p \rangle \qquad (i = 1, 2, \ldots). \tag{7'}$$

The necessary condition for the graph G to have a nontrivial automorphism group $\mathcal{G}$ is hence the existence of vertices s and p which satisfy (7').

Using nondiagonal elements of $\boldsymbol{K}_i$ and $\boldsymbol{K}_i'$, one can obtain another set of necessary conditions. Consider matrix elements $\langle s \mid \boldsymbol{K}_i \mid p \rangle$ ($p = 1, \ldots, n$) which form the sth row of a projection operator $\boldsymbol{K}_i$. There exists a permutation $\mathbf{P}$ which permutes vertices (p) in such a way that matrix elements $\langle s \mid \boldsymbol{K}_i \mid \mathbf{P}p \rangle$ ($p = 1, \ldots, n$) form a descending sequence. Let $\mathcal{R}(\boldsymbol{K}_i, s)$ denote this sequence. For each projection operator $\boldsymbol{K}_i$ there are n such descending sequences. These sequences can be also ordered in descending sequence; let $\mathcal{R}(\boldsymbol{K}_i)$ denote a descending sequence of such sequences. One again finds that the sequence $\mathcal{R}(\boldsymbol{K}_i)$ does not depend on the particular labeling of graph vertices. Hence, and from theorem 1, one can derive:

COROLLARY 2

A necessary condition for graphs G and G' to be isomorphic is that

$$\mathcal{R}(\boldsymbol{K}_i) = \mathcal{R}(\boldsymbol{K}_i') \qquad (i = 1, 2, \ldots). \tag{8}$$

Corollaries 1 and 2 are powerful practical tests for graph isomorphism. These corollaries do not solve the graph isomorphism problem, but they eliminate a very large number of graphs which are isospectral but not isomorphic. If graphs are not isomorphic, then both tests usually fail already in the case of the first projection

operators $\boldsymbol{K}_1$ and $\boldsymbol{K}'_1$. This substantially decreases the average operation count needed to perform these tests.

Conditions (7) and (8) are not sufficient for graph isomorphism, since these conditions allow for the unrestricted use of permutations **P**. Condition (5), which in conjuncture with the requirement $\lambda_i = \lambda'_i$ is also sufficient, requires that *the same* permutation **P** should be used in the case of all projection operators $\boldsymbol{K}_i$. Conditions (7) and (8) should hence be modified in such a way as to guarantee the existence of a *unique* permutation **P** performing a desired transformation. With an appropriate modification and some additional bookkeeping, these conditions (in conjuncture with the isospectrality requirement) can also become sufficient.

Let us now return to the problem of the unique graph ID. We will present here only the general idea of the algorithm. This algorithm is based on the construction of the set $\mathbb{S}$ which contains all "canonical" labelings of the graph G.

In order that the set $\mathbb{S}$ can be used for a construction of a unique graph ID, we require this set to have the following properties. First, the elements $\mathcal{S} \in \mathbb{S}$ of $\mathbb{S}$ are different labelings of a graph G. The set $\mathbb{S}$ as a whole is a graph invariant in the sense that it is independent of the particular initial labeling of G. In other words, each adjacency matrix **A** of G should produce the same set $\mathbb{S}$. Further, if $\mathcal{S} \in \mathbb{S}$ is an element of $\mathbb{S}$ and if $\mathbf{P} \in \mathcal{G}$ is an element of the graph automorphism group $\mathcal{G}$, then $\mathbf{P}\mathcal{S} = \mathcal{S}' \in \mathbb{S}$ is another element of $\mathbb{S}$. Finally, if $\mathcal{S} \in \mathbb{S}$ and $\mathcal{S}' \in \mathbb{S}$ are two elements of $\mathbb{S}$, then there exists a permutation $\mathbf{P} \in \mathcal{G}$ such that $\mathbf{P}\mathcal{S} = \mathcal{S}'$. In short, the set $\mathbb{S}$ is a graph invariant and in addition it is invariant and irreducible with respect to the automorphism group $\mathcal{G}$.

We call a set $\mathbb{S}$ with the above properties a "canonical" set and we call each element $\mathcal{S}$ of $\mathbb{S}$ a "canonical" labeling. This definition of a canonical set is much stronger than the definition usually considered in the literature.

The above definition of the set $\mathbb{S}$ guarantees the construction of the unique graph ID. This construction can be done in the following way. Since $\mathbb{S}$ is invariant and irreducible with respect to $\mathcal{G}$, the adjacency matrix **A** constructed according to the canonical labeling $\mathcal{S} \in \mathbb{S}$ does not depend on a particular choice of $\mathcal{S}$. Such an adjacency matrix is hence a function of the entire set $\mathbb{S}$, and not of the particular element of $\mathbb{S}$. One can write the elements of the upper triangle of **A** as a continuous string of zeros and ones. Each three digits in this string can then be replaced with a single octal number. The resulting string thus derived is a graph invariant. Furthermore, this string uniquely defines the adjacency matrix **A** and hence graph G. Therefore, one can use this string as a unique graph ID.

In addition to a construction of a unique graph ID, the set $\mathbb{S}$ also solves the graph automorphism problem. Each element $\mathcal{S} \in \mathbb{S}$ is an equivalent canonical labeling of G, and all these elements are mutually connected by the elements of the automorphism group $\mathcal{G}$. In particular, if $\mathbb{S}$ contains only one element, the group $\mathcal{G}$ is trivial, i.e. it contains only the identity.

The above definition of the set $\mathbb{S}$ is temporarily a statement of the desirable properties of this set. The main problem is the construction of a set $\mathbb{S}$ with such

properties. We construct this set in an iterative way by forming a sequence of the intermediate sets $\mathbb{S}_i$ ($i = 0, 1, \ldots$). The last element of this sequence is required to be the set $\mathbb{S}$. Each set $\mathbb{S}_k$ contains all up to this point *admissable* labelings of G. In order to form these intermediate sets, we use projection operators $\boldsymbol{K}_i$ of the spectral decomposition (2). These projection operators are considered one by one according to the descending sequence of the corresponding eigenvalues λ_i. Not all admissable labelings contained in the set $\mathbb{S}_i$ are the final canonical labelings, and as the algorithm proceeds, the number of admissable labelings decreases. Initially, all $n!$ labelings of the graph G are admissable, and hence the set $\mathbb{S}_0$ contains $n!$ elements. In the ith step of the algorithm, the current projection operator $\boldsymbol{K}_i$ eliminates some labelings contained in the set $\mathbb{S}_{i-1}$. Each set $\mathbb{S}_i$ is hence a subset of the previous set $\mathbb{S}_{i-1}$, and the final set contains only canonical labelings.

We also require that the kth set $\mathbb{S}_k$ satisfies the following lemma:

LEMMA 1

Let $\mathcal{S} \in \mathbb{S}_k$ and $\mathcal{S}' \in \mathbb{S}_k$ be elements of the set $\mathbb{S}_k$, and let $\mathbf{P}$ be a permutation which transforms labeling $\mathcal{S}$ into labeling $\mathcal{S}'$, i.e. let $\mathbf{P}\mathcal{S} = \mathcal{S}'$. Then,

$$\langle s \mid \boldsymbol{K}_i \mid p \rangle = \langle \mathbf{P}s \mid \boldsymbol{K}_i \mid \mathbf{P}p \rangle, \qquad (s, p = 1, \ldots, n) \tag{9}$$

for all projection operators $\boldsymbol{K}_i$ ($i = 1, \ldots, k$). Inversely, if the permutation $\mathbf{P}$ satisfies relation (9) for all projection operators $\boldsymbol{K}_i$ ($i = 1, \ldots, k$), and if $\mathcal{S} \in \mathbb{S}_k$, then $\mathcal{S}' = \mathbf{P}\mathcal{S} \in \mathbb{S}_k$.

The above lemma is a statement of desirable properties of the set $\mathbb{S}_k$, and we will construct these sets in such a way as to satisfy this lemma. According to theorem 1, such a construction guarantees that the sequence of intermediate sets $\mathbb{S}_i$ terminates with the set $\mathbb{S}$ with the required properties.

It is convenient to represent the set $\mathbb{S}_k$ as a collection of sets $\mathcal{S}_\nu = \{\mathbf{L}_1^\nu, \mathbf{L}_2^\nu, \ldots, \mathbf{L}_r^\nu\}$ ($\nu = 1, 2, \ldots, \mathcal{N}$). Each set $\mathcal{S}_\nu$ contains a descending sequence of equivalence classes $\mathbf{L}_i^\nu$. These classes contain vertices of G which are up to this point equivalent. Obviously, $\sum_i N_i = n$, where N_i is the number of vertices in the class $\mathbf{L}_i^\nu$. The set $\mathcal{S}_\nu$ defines a class of admissable labelings of G. The first N_1 vertices $(s) \in \mathbf{L}_1^\nu$ can be labeled with the first N_1 labels, the next N_2 vertices $(s) \in \mathbf{L}_2^\nu$ can be labeled with the next N_2 labels, etc. Hence, each set $\mathcal{S}_\nu$ defines $N_1!\,N_2! \ldots N_r!$ admissable labelings of G. It is more convenient to work with such sets instead of with the particular admissable labelings. For the sake of simplicity, we will denote the set $\mathcal{S}_\nu$, the class of the corresponding admissable labelings, as well as each particular labeling $\mathcal{S} \in \mathbb{S}_k$ with the same symbol. Strictly, one should distinguish between all these quantities.

Assuming that lemma 1 is true, one finds that all $\mathcal{N}$ sets $\mathcal{S}_\nu = \{\mathbf{L}_1^\nu, \mathbf{L}_2^\nu, \ldots, \mathbf{L}_r^\nu\} \in \mathbb{S}_k$ are mutually isomorphic, i.e. they contain the same number of equivalence classes K_i and the same number N_i of vertices contained in these

classes. Since each set $\mathcal{S}_v$ contains $N_1!N_2!\ldots N_r!$ admissable labelings, the set $\mathbb{S}_k$ contains $\mathcal{N}N_1!N_2!\ldots N_r!$ admissable labelings of the graph G.

Since initially all $n!$ labelings of G are admissable, the set $\mathbb{S}_0$ contains a single element $\mathcal{S}=\{\mathbf{L}\}$, where the equivalence class $\mathbf{L}$ contains all vertices of the graph G. As the algorithm proceeds, each current set $\mathcal{S}_v$ can split into one or more subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots$ and each class $\mathbf{L}_i \in \mathcal{S}_v$ can split into two or more subclasses $\mathbf{L}_1, \mathbf{L}_2, \ldots$, etc.

Let us now see in more detail how this is done. We will first consider the case when the eigenvalue λ_k is nondegenerate, and next we will consider, in somewhat less detail, the case when λ_k is degenerate.

Let λ_k be nondegenerate, and let the set $\mathbb{S}_{k-1}$ satisfy lemma 1. We now show how the projection operator $\boldsymbol{K}_k$ generates the set $\mathbb{S}_k$ which is the subset of the set $\mathbb{S}_{k-1}$ and which also satisfies lemma 1.

Since λ_k is nondegenerate, matrix elements of the projection operator $\boldsymbol{K}_k$ are

$$\langle s \mid \boldsymbol{K}_k \mid p\rangle = \langle s \mid \Psi_k\rangle \langle \Psi_k \mid p\rangle. \tag{10}$$

We now form the descending sequence of the coefficients $c_s \equiv \langle s \mid \Psi_k\rangle$ and the descending sequence of the coefficients $-c_s \equiv -\langle s \mid \Psi_k\rangle$. This latter sequence is exactly the inverse of the former. Next, we compare lexicographically these two sequences. If these sequences are not equal, we choose the larger sequence. Otherwise, both sequences are retained.

Assume first that the descending sequence of the coefficients c_s is lexicographically the dominant one, and let the set $\mathbb{S}_{k-1}$ contain $\mathcal{N}_{k-1}$ sets $\mathcal{S}_v = \{\mathbf{L}_1^v, \mathbf{L}_2^v, \ldots, \mathbf{L}_r^v\}$. By assumption, all these sets are mutually isomorphic. Each set $\mathcal{S}_v$ in conjuncture with the eigenvector Ψ_k generates an ordered set $\mathcal{F}(\mathcal{S}_v, \Psi_k)$ of the coefficients c_s. The first N_1 elements of $\mathcal{F}(\mathcal{S}_v, \Psi_k)$ are the coefficients c_s $(s \in \mathbf{L}_1^v)$ taken in descending order. The next N_2 elements of $\mathcal{F}(\mathcal{S}_v, \Psi_k)$ are the coefficients c_s $(s \in \mathbf{L}_2^v)$ taken in descending order, etc. We compare lexicographically all $\mathcal{N}_{k-1}$ sets $\mathcal{F}(\mathcal{S}_v, \Psi_k)$ obtained in this way and retain only $\mathcal{N}' \leq \mathcal{N}_{k-1}$ of these sets which are lexicographically largest and mutually isomorphic. We eliminate all sets $\mathcal{S}_v$ associated with the excluded $(\mathcal{N}_{k-1} - \mathcal{N}')$ sets $\mathcal{F}(\mathcal{S}_v, \Psi_k)$.

The remaining $\mathcal{N}'$ sets $\mathcal{F}(\mathcal{S}_v, \Psi_k)$ are used in order to update the original sets $\mathcal{S}_v \in \mathbb{S}_{k-1}$ into new sets $\mathcal{S}_v' \in \mathbb{S}_k$. If vertices (s) and (p) which are initially contained in the same class $\mathbf{L}_i^v \in \mathcal{S}_v$ satisfy $c_s > c_p$, then the class $\mathbf{L}_i^v$ splits into two subclasses $\mathbf{L}_{i1}^v$ and $\mathbf{L}_{i2}^v$ such that $(s) \in \mathbf{L}_{i1}^v$, $(p) \in \mathbf{L}_{i2}^v$ and, in addition, $\mathbf{L}_{i1}^v > \mathbf{L}_{i2}^v$. In this way, the number of classes contained in the set $\mathcal{S}_v$ increases, and the number of admissable labelings of G decreases. Since the original $\mathcal{N}_{k-1}$ sets $\mathcal{S}_v$ are by assumption isomorphic and since the $\mathcal{N}'$ sets $\mathcal{F}(\mathcal{S}_v, \Psi_k)$ considered are also isomorphic, the final $\mathcal{N}'$ sets $\mathcal{S}_v$ are mutually isomorphic as well.

As a result, the original set $\mathbb{S}_{k-1}$ containing $\mathcal{N}_{k-1}$ mutually isomorphic sets $\mathcal{S}_v$ changes into the set $\mathbb{S}_k$ containing $\mathcal{N}' = \mathcal{N}_k$ mutually isomorphic sets $\mathcal{S}_v'$. It can be shown that the set $\mathbb{S}_k$ satisfies lemma 1. This follows from the fact that the set

$\mathbb{S}_k$ is a subset of a set $\mathbb{S}_{k-1}$ (this can be easily seen from the above construction), from the assumption that the set $\mathbb{S}_{k-1}$ satisfies this lemma, and from the fact that all sets $\mathcal{F}(\mathcal{S}_\nu, \Psi_k)$ which are used in order to update particular sets $\mathcal{S}_\nu$ are isomorphic [25].

If the descending sequence of coefficients c_s equals the descending sequence of inverse coefficients $-c_s$, both possibilities are considered. In this case, each set $\mathcal{S}_\nu \in \mathbb{S}_{k-1}$ in conjuncture with the eigenvector Ψ_k generates two sets $\mathcal{F}_{\nu 1}$ and $\mathcal{F}_{\nu 2}$. One again retains only lexicographically dominant and mutually isomorphic sets $\mathcal{F}_{\nu 1}$ and $\mathcal{F}_{\nu 2}$. In particular, if for some ν both sets $\mathcal{F}_{\nu 1}$ and $\mathcal{F}_{\nu 2}$ are retained, the orginal set $\mathcal{S}_\nu$ splits into two sets $\mathcal{S}_{\nu 1}$ and $\mathcal{S}_{\nu 2}$. In this way, the set $\mathbb{S}_{k-1}$ changes into a new set $\mathbb{S}_k$. One again shows that the set $\mathbb{S}_k$ is a subset of the set $\mathbb{S}_{k-1}$, and that it satisfies lemma 1 [25].

If all eigenvalues λ_i of the adjacency matrix **A** are nondegenerate, the above method leads to the unique graph ID, and to the solution of the graph isomorphism problem. Since there is no projection operator $\boldsymbol{K}_0$, the initial set $\mathbb{S}_0$ trivially satisfies lemma 1. In addition, each set $\mathbb{S}_k$ is a subset of a previous set $\mathbb{S}_{k-1}$. One now shows by induction that all sets $\mathbb{S}_k$ satisfy this lemma. In particular, the last set in this sequence satisfies relation (9) for all projection operators $\boldsymbol{K}_i$. According to theorem 2, this implies that this set is invariant and irreducible with respect to the automorphism group $\mathcal{G}$, i.e. this is the required set $\mathbb{S}$.

It remains to generalize this approach to degenerate eigenvalues λ_i. If λ_k is μ-degenerate, the projection operator $\boldsymbol{K}_k$ is given by a more general expression (3a), and degenerate eigenvectors $|\Psi_{k\nu}\rangle$ ($\nu = 1, \ldots, \mu$) are ambiguous up to the unitary transformation in the corresponding μ-dimensional subspace. According to (3a),

$$\langle s|\boldsymbol{K}_k|p\rangle = \sum_{\nu}^{\mu} \langle s|\Psi_{k\nu}\rangle\langle\Psi_{k\nu}|p\rangle = (s|p), \tag{11}$$

where $|s)$ is a μ-dimensional vector with components $\langle\Psi_{k\nu}|s\rangle$, and where $(s|p)$ is the scalar product between two such vectors. Matrix elements of the projection operator $\boldsymbol{K}_k$ are thus scalar products between μ-dimensional vectors $|s)$.

The main idea of the approach is the same as in the nondegenerate case. However, in order to update the current set $\mathcal{S}_\nu \in \mathbb{S}_{k-1}$, we can no longer use components $\langle s|\Psi_{k\nu}\rangle$ of the eigenvectors $|\Psi_{k\nu}\rangle$, since these eigenvectors are ambiguous up to the unitary transformation in the corresponding degenerate subspace. Instead, we use scalar products $(s|p)$.

With each vertex (s), we associate an ordered set $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$ of scalar products $(s|p)$ ($p = 1, \ldots, n$). This set is ordered using two keys. As a primary key we use the set $\mathcal{S}_\nu$, and as a secondary key we use the descending sequence of the scalar products $(s|p)$. In other words, the first N_1 elements of $\mathcal{F}$ are scalar products $(s|p)$, $(p) \in \boldsymbol{K}_1^\nu$ taken in descending order, the next N_2 elements of $\mathcal{F}$ are scalar products $(s|p)$, $(p) \in \boldsymbol{K}_2^\nu$ taken in descending order, etc.

Next, we form a descending sequence $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ of such sets. This is done using three keys. As a primary key we use the set $\mathcal{S}_\nu$, as a secondary key we use the norm $(s \mid s)$ of a vector $\mid s)$, and as a tertiary key we use the lexicographical order of sets $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$. In other words, the first N_1 elements of $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ are N_1 sets $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$, $(s) \in \boldsymbol{K}_1^\nu$. These sets are ordered according to the descending sequence of the corresponding norms $(s \mid s)$. Finally, if some sets $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$, $(s) \in \boldsymbol{K}_1^\nu$ have the same associated norm $(s \mid s)$, then these sets are ordered lexicographically. Similarly for the following N_2 elements of $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$, etc.

We compare all sets $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ $(\nu = 1, 2, \ldots)$ constructed in this way and retain only those sets which are lexicographically dominant and mutually isomorphic [25]. Next, we update sets $\mathcal{S}_\nu$. The sets $\mathcal{S}_\nu$ associated with the excluded sets $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ are not updated and they are ignored; only the sets $\mathcal{S}_\nu$ associated with the remaining sets $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ are updated.

Let $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ be one such set. One first considers the leading element $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$ of $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$. This element is an ordered set of the scalar products $(p \mid s)$ $(p = 1, \ldots, n)$. This set changes the original set $\mathcal{S}_\nu$ into a new set $\mathcal{S}_\nu'$ in a similar way as in the nondegenerate case described above. If the leading term of $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ is not unique, i.e. if there are more than one isomorphic leading elements of $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$, each of these elements generates the corresponding set $\mathcal{S}_\nu'$. Next, one considers the following element of $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ in order to further update the obtained sets $\mathcal{S}_\nu'$, etc. This process terminates with the update of the set $\mathcal{S}_\nu$ to one or more sets S_ν'.

In the above way, the set $\mathbb{S}_{k-1}$ in conjuncture with the projection operator $\boldsymbol{K}_k$ generates a new set $\mathbb{S}_k$. How this is done is illustrated with one example in the next section, and it will be described in more detail elsewhere [25]. The set $\mathbb{S}_k$ constructed in this way can again be shown to satisfy lemma 1, and the whole iterative process hence terminates with the set $\mathbb{S}$

Concerning the above method, two comments are needed. First, concerning the treatment of the degenerate case, note that in order to fix n^2 scalar products $(s \mid p)$, much less scalar products are needed. This can substantially decrease the required operation count of the algorithm. For example, in the case $\mu = 2$, all n vectors $\mid s)$ are two-dimensional, and hence these vectors lie in the same plane. All n^2 scalar products $(s \mid p)$ $(s, p = 1, \ldots, n)$ are hence uniquely fixed with only $2n$ scalar products $(s_1 \mid p)$ and $(s_2 \mid p)$ $(p = 1, \ldots, n)$, provided vectors $\mid s_1)$ and $\mid s_2)$ are linearly independent. This follows from the fact that $(s_1 \mid p)$ and $(s_2 \mid p)$ can be considered as two coordinates of a vector $\mid p)$, and since this vector is two-dimensional, it is fixed with these coordinates.

More generally, if λ_k is μ-degenerate, only μn scalar products $(s \mid p)$ are absolutely required in order to fix all n^2 scalar products. Accordingly, it is not necessary to consider all elements $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$ of the set $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k)$ in order to update the set $\mathcal{S}_\nu'$. It is only necessary to consider μ leading elements, provided the corresponding μ vectors $\mid s)$ are linearly independent. There is a slight complication if any of the excluded elements $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$ are isomorphic with an included element $\mathcal{F}(\mathcal{S}_\nu, \boldsymbol{K}_k, s)$.

In this case, such excluded elements should also be retained in the set $\mathcal{F}(S_v, \boldsymbol{K}_k)$. This is necessary in order to treat all admissable labelings of G in an impartial way, i.e. in order to guarantee that the set $\mathbb{S}_k$ is graph invariant [25].

The second comment considers the termination of the algorithm. In many cases, it is not necessary to consider all projection operators $\boldsymbol{K}_i$. This is particularly true in the case of the graph automorphism problem. Namely, if at any point of the algorithm the number of admissable labelings $S \in \mathbb{S}_k$ is reasonably small, one can determine the automorphism group G using directly relation (1'). In addition, since graph ID is not required, it is not necessary to consider projection operators $\boldsymbol{K}_k$ in any particular sequence. This flexibility can substantially reduce the operation count of the algorithm.

With some modification, the above approach can also be applied to the problem of graph isomorphism. In general, the graph isomorphism problem can be treated in the following way.

Let G and G' be graphs and let $\mathbf{A}$ and $\mathbf{A}'$ be the corresponding adjacency matrices. We first diagonalize $\mathbf{A}$ and $\mathbf{A}'$. Next, we compare $\mathbf{A}$ and $\mathbf{A}'$ for isospectrality. If these matrices are not isospectral, graphs G and G' are not isomorphic, and the algorithm is terminated. If, however, $\mathbf{A}$ and $\mathbf{A}'$ are isospectral, the decision is not yet obtained, and the search for the possible isomorphism should be continued. Here we use theorem 1 and in particular relation (5).

We examine each pair $\{\boldsymbol{K}_i, \boldsymbol{K}_i'\}$ $(i = 1, 2, \ldots)$ for the condition (5). The examination can in principle proceed in any sequence. This flexibility can be used in order to improve the numerical performance of the algorithm. If for any pair $\{\boldsymbol{K}_k, \boldsymbol{K}_k'\}$ the test (5) fails, graphs G and G' are not isomorphic, and the algorithm is terminated.

In order to verify relation (5), we compare ordered sets $\mathcal{F}(S_v, \Psi_k)$ and $\mathcal{F}(S_v', \Psi_k')$ in the case when the eigenvalue λ_k is nondegenerate, and we compare ordered sets $\mathcal{F}(S_v, \boldsymbol{K}_k)$ and $\mathcal{F}(S_v', \boldsymbol{K}_k')$ in the case when the eigenvalue λ_k is degenerate. If the comparison fails, this implies that there is no permutation $\mathbf{P}$ which satisfies (5) for $i = 1, \ldots, k$. The algorithm hence terminates, and graphs G and G' are found not to be isomorphic. Otherwise, we update all sets S_v and all sets S_v', and we proceed with the next pair $\{\boldsymbol{K}_{k+1}, \boldsymbol{K}_{k+1}'\}$ of projection operators.

It is numerically advantageous to first consider nondegenerate eigenvalues λ_i, then doubly degenerate eigenvalues λ_i, etc. This is due to the fact that it is much simpler to form sets S_v which are generated by projection operators $\boldsymbol{K}_i$ which correspond to nondegenerate eigenvalues, than to form sets S_v which are generated by projection operators $\boldsymbol{K}_i$ which correspond to degenerate eigenvalues. When all nondegenerate eigenvalues λ_i are exhausted, the number of admissable labelings is usually drastically reduced. If this is the case, one can terminate the algorithm with a direct vertification of relation (1) instead of proceeding with the remaining projection operators. In most cases, this approach can substantially reduce the operation count of the algorithm.

3. An example

In order to illustrate the above algorithms, let us consider an example. First, we will consider a graph isomorphism problem.

In fig. 1 are shown graphs G and G'. We have intentionally chosen two isomorphic graphs. This is the worst possible case for the problem of graph isomorphism. If the graphs are not isomorphic, the algorithm usually terminates much faster, and the corresponding operational count is much smaller.

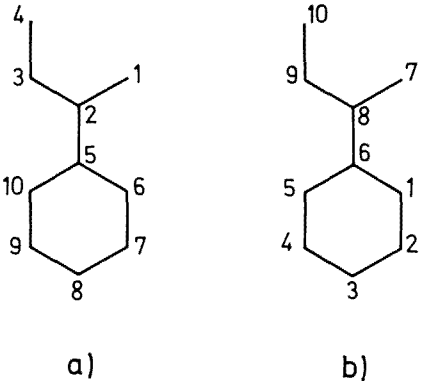

Fig. 1. Two isomorphic graphs G and G'.

Graphs G and G' contain 10 vertices each, and hence one can label each of these graphs in 10! = 3,628,800 different ways. A particular choice of labels for G and G' is shown in fig. 1. Accordingly, we have two adjacency matrices, **A** and **A'**.

First, one diagonalizes **A** and **A'**. One finds that **A** and **A'** are isospectral with the eigenvalues

$$\{2.2143,\ 1.6751,\ 1,\ 1,\ 0.5392,\ -0.5392,\ -1,\ -1,\ -1.6751,\ -2.2143\}. \quad (12)$$

Hence, one has to continue the algorithm. According to (12), there are eight projection operators $\boldsymbol{K}_i$ and eight projection operators $\boldsymbol{K}_i'$. Six pairs $\{\boldsymbol{K}_i, \boldsymbol{K}_i'\}$ correspond to the nondegenerate eigenvalues, and two correspond to the doubly degenerate eigenvalues $\lambda = 1$ and $\lambda = -1$.

Consider first the pair $\{\boldsymbol{K}_1, \boldsymbol{K}_1'\}$ which corresponds to the largest eigenvalue $\lambda_1 = 2.2143$. The corresponding nondegenerate normalized eigenvectors are

$$\begin{aligned}\Psi_1 &= 0.1928\Phi_1 + 0.4269\Phi_2 + 0.2422\Phi_3 + 0.1094\Phi_4 + 0.5103\Phi_5 \\ &\quad + 0.3515\Phi_6 + 0.2681\Phi_7 \pm 0.2422\Phi_8 + 0.2681\Phi_9 + 0.3515\Phi_{10};\end{aligned}$$

$$\begin{aligned}\Psi_1' &= 0.3515\Phi_1' + 0.2681\Phi_2' + 0.2422\Phi_3' + 0.2681\Phi_4' + 0.3515\Phi_5' \\ &\quad + 0.5103\Phi_6' + 0.1928\Phi_7' + 0.4269\Phi_8' + 0.2422\Phi_9' + 0.1094\Phi_{10}', \qquad (13)\end{aligned}$$

where Φ_i is the state associated with the vertex (i) of the graph G, while Φ_i' is the state associated with the vertex (i) of the graph G'. The sign of Ψ_1 is fixed, with the requirement that the coefficient $c_5 = 0.5103$, which is in absolute value the largest, should be positive. In a similar way, the sign of Ψ_1' is fixed.

Initially, all 10! labelings of G as well as all 10! labelings of G' are admissable. Hence, $\mathbb{S}_0 \ni \mathcal{S} = \{\mathbf{L}\}$ and $\mathbb{S}_0' \ni \mathcal{S}' = \{\mathbf{L}'\}$, where the set $\mathbf{L}$ contains all vertices of G, while the set $\mathbf{L}'$ contains all vertices of G'. One easily finds that the ordered set $\mathcal{F}(\mathcal{S}, \Psi_1)$ generated from the coefficients $c_s = \langle s \mid \Psi_1 \rangle$ of the eigenvector Ψ_1 equals the ordered set $\mathcal{F}'(\mathcal{S}', \Psi_1')$ generated from the coefficients $c_s' = \langle s \mid \Psi_1' \rangle$ of the eigenvector Ψ_1'. Hence, the pair $\{\boldsymbol{K}_1, \boldsymbol{K}_1'\}$ satisfies relation (5), and one has to continue the algorithm.

Next, one constructs sets $\mathbb{S}_1$ and $\mathbb{S}_1'$. From (13), one easily finds $\mathbb{S}_1 \equiv \{\mathcal{S}\}$ and $\mathbb{S}_1' \equiv \{\mathcal{S}'\}$, where

$$\mathcal{S} \equiv \{\mathbf{L}_i\} = \{(5)\,(2)\,(6, 10)\,(7, 9)\,(3, 8)\,(1)\,(4)\},$$

$$\mathcal{S}' \equiv \{\mathbf{L}_i'\} = \{(6)\,(8)\,(1, 5)\,(2, 4)\,(9, 3)\,(7)\,(19)\}. \tag{14a}$$

Each admissable canonical labeling of graph G should hence assign label 1 to the vertex (5), label 2 to the vertex (2), labels 3 and 4 to the class $\mathbf{L}_3 = (6, 10)$, labels 5 and 6 to the class $\mathbf{L}_4 = (7, 9)$, labels 7 and 8 to the class $\mathbf{L}_5 = (3, 8)$, label 9 to the vertex (1), and label 10 to the vertex (4). Similarly, each admissable labeling of the graph G' should assign label 1 to the vertex (6), label 2 to the vertex (8), labels 3 and 4 to the class $\mathbf{L}_3' = (1, 5)$, etc. If graphs G and G' are isomorphic, then vertices (5), (2), (1) and (4) in the graph G should correspond to vertices (6), (8), (7) and (10) in the graph G', respectively. Similarly, vertices (6) and (10) in G should correspond to vertices (1) and (5) in G', vertices (7) and (9) in G should correspond to vertices (2) and (4) in G', while vertices (3) and (8) in G should correspond to vertices (9) and (3) in G'. The initial 10! admissable labelings are hence reduced to only 2! 2! 2! = 8 admissable labelings. This illustrates how the test (5) performed on a single projection operator $\boldsymbol{K}_i$ can dramatically reduce the number of admissable permutations and graph labelings.

Next we proceed with the normalized eigenvectors Ψ_2 and Ψ_2' which correspond to the nondegenerate eigenvalue $\lambda_2 = 1.6751$:

$$\begin{aligned}\Psi_2 = {} & 0.2699\Phi_1 + 0.4521\Phi_2 + 0.4193\Phi_3 + 0.2503\Phi_4 + 0.0681\Phi_5 \\ & - 0.1690\Phi_6 - 0.3512\Phi_7 - 0.4193\Phi_8 - 0.3512\Phi_9 - 0.1690\Phi_{10};\end{aligned}$$

$$\begin{aligned}\Psi_2' = {} & 0.1690\Phi_1' + 0.3512\Phi_2' + 0.4193\Phi_3' + 0.3512\Phi_4' + 0.1690\Phi_5' \\ & - 0.0681\Phi_6' - 0.2699\Phi_7' - 0.4521\Phi_8' - 0.4193\Phi_9' - 0.2503\Phi_{10}'.\end{aligned} \tag{15}$$

Since $c_6 = c_{10}$ and $c_7 = c_9$, the coefficients of Ψ_2 do not discriminate between vertices contained in classes $\mathbf{L}_3$ and $\mathbf{L}_4$, respectively. However, $c_3 > c_8$, and hence these coefficients discriminate between vertices (3) and (8) contained in class $\mathbf{L}_5$. This class splits into two subclasses according to $\mathbf{L}_5 = (3, 8) \Rightarrow (3)\ (8)$. Labels 7 and 8 should therefore be assigned, respectively, to vertices (3) and (8) of the graph G.

One similarly finds that the coefficients c_i' of the eigenvector Ψ_2' do not discriminate between vertices in classes $\mathbf{L}_3'$ and $\mathbf{L}_4'$. However, these coefficients discriminate between vertices (9) and (3) of class $\mathbf{L}_5'$. One thus finds that the labels 7 and 8 should be assigned to vertices (9) and (3) of the graph G'. If G and G' are isomorphic, vertices (3) and (8) in graph G should correspond, respectively, to vertices (9) and (3) of graph G'. Hence, $\mathbb{S}_2 \equiv \{S\}$ and $\mathbb{S}_2' \equiv \{S'\}$, where the updated sets S and S' are

$$S \equiv \{\mathbf{L}_i\} = \{(5)\ (2)\ (6, 10)\ (7, 9)\ (3)\ (8)\ (1)\ (4)\},$$

$$S' \equiv \{\mathbf{L}_i'\} = \{(6)\ (8)\ (1, 5)\ (2, 4)\ (9)\ (3)\ (7)\ (10)\}. \tag{14b}$$

The number of admissable labelings is now reduced to only four. At this point, as well as already at the previous one, one could directly check for graph isomorphism using relation (1). One has to verify this relation for only four admissable labelings, which is numerically very efficient. However, we proceed with the basic algorithm in order to demonstrate how it works in the case of degenerate eigenvalues.

Next we consider the doubly degenerate eigenvalue $\lambda_3 = 1$ and the corresponding pair $\{\boldsymbol{K}_3, \boldsymbol{K}_3'\}$ of projection operators. Up to this point, there are only two classes which are left ambiguous, classes $\mathbf{L}_3 = (6, 10)$ and $\mathbf{L}_4 = (7, 9)$ in the case of the graph G, and the corresponding classes $\mathbf{L}_3' = (1, 5)$ and $\mathbf{L}_4' = (2, 4)$ in the case of the graph G'.

Since λ_3 is doubly degenerate, vectors $|s)$ which define scalar products $(s\,|\,p) = \langle s\,|\,\boldsymbol{K}_3\,|\,p\rangle$ are two-dimensional, and hence they can be drawn in a single plane. These vectors are shown in fig. 2. The current set S fixes the sequence of all vertices (s), except the relative sequence of vertices (6) and (10), and the relative sequence of vertices (7) and (9). Hence, we use the projection operator $\boldsymbol{K}_3$ in order to determine, if possible, these relative sequences.

First we form a descending sequence of sets $\mathcal{F}(S, \boldsymbol{K}_3, s)$. As a primary key for this sequence we use a set S, as a secondary key we use the norm $(s\,|\,s)$ of a vector $|s)$, and as a tertiary key we use a lexicographical order of such sets. According to the discussion in the previous section, it is not necessary to consider all sets $\mathcal{F}(S, \boldsymbol{K}_3, s)$. Since vectors $|s)$ are two-dimensional, one has to consider only those leading sets of $\mathcal{F}(S, \boldsymbol{K}_3, s)$ which form a graph invariant set $\mathcal{F}(S, \boldsymbol{K}_3)$ and which involve at least two linearly independent vectors. This is done in the following way.

According to the primary key, the first element of the set $\mathcal{F}(S, \boldsymbol{K}_3)$ is $\mathcal{F}(S, \boldsymbol{K}_3, 5)$, the second element of this set is $\mathcal{F}(S, \boldsymbol{K}_3, 2)$, the next two elements are $\mathcal{F}(S, \boldsymbol{K}_3, 6)$

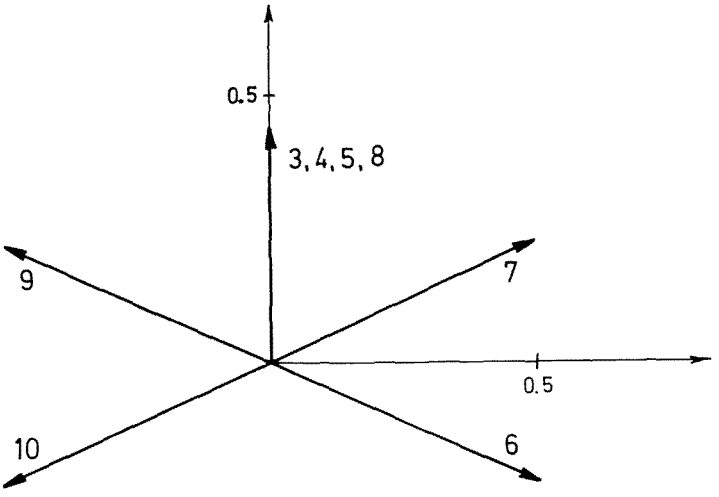

Fig. 2. Vectors $|s)$ associated with the doubly degenerate eigenvalue $\lambda_3 = 1$. Graph vertices are labeled as shown in fig. 1(a).

and $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 10)$, etc. Since $(2|2) = 0$, vector $|2)$ is linearly dependent on vector $|5)$, and hence one has to go beyond the second element in $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3)$. One finds that the next two elements $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 6)$ and $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 10)$ are isomorphic, and hence one has to include both these elements. Since the vector $|6)$ (as well as the vector $|10)$) is linearly independent of the vector $|5)$, it is not necessary to consider further elements in the set $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3)$.

Next we consider the above four sets $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, s)$. Each $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, s)$ is an ordered set of scalar products $(p|s)$ $(p = 1, \ldots, 10)$. As a primary key we use the set $\mathcal{S}$, and as a secondary key we use a descending sequence of the scalar products $(p|s)$ $(p = 1, \ldots, n)$. Hence, the first element of the set $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, s)$ is the scalar product $(5|s)$, the next element of this set is the scalar product $(2|s)$, the next two elements are scalar products $(6|s)$ and $(10|s)$ in descending order, etc.

One finds that the first set $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 5)$ does not discriminate between vertices (6, 10) and (7, 9). This is geometrically obvious from fig. 2. Since $|2) = 0$, also the second set $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 2)$ does not discriminate between these vertices. One further finds $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 6) = \mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 10)$. Hence, both sets should be used in order to update the set $\mathcal{S}$. Since $(7|6) > (9|6)$, the set $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 6)$ updates $\mathcal{S}$ in such a way that $\mathbf{L}_3 = (6, 10) \Rightarrow (6)\ (10)$ and $\mathbf{L}_4 = (7, 9) \Rightarrow (7)\ (9)$. Similarly, since $(10|9) > (10|7)$, the set $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 10)$ updates $\mathcal{S}$ in such a way that $\mathbf{L}_3 = (6, 10) \Rightarrow (10)\ (6)$ and $\mathbf{L}_4 = (7, 9) \Rightarrow (9)\ (7)$. Hence, $\mathcal{S}$ splits into two isomorphic sets $\mathcal{S}_1$ and $\mathcal{S}_2$. In other words, $\mathbb{S}_3 \equiv \{\mathcal{S}_1, \mathcal{S}_2\}$, where

$$\mathcal{S}_1 \equiv \{\mathbf{L}_i^1\} = \{(5)\,(2)\,(6)\,(10)\,(7)\,(9)\,(3)\,(8)\,(1)\,(4)\},$$

$$\mathcal{S}_2 \equiv \{\mathbf{L}_i^2\} = \{(5)\,(2)\,(10)\,(6)\,(9)\,(7)\,(3)\,(8)\,(1)\,(4)\}. \tag{14c}$$

Hence, there are only two admissable labelings for a graph G.

In a similar way, the projection operator $\boldsymbol{K}_3'$ associated with the graph G' generates corresponding sets $\mathcal{F}'(\mathcal{S}', \boldsymbol{K}_3', 6)$, $\mathcal{F}'(\mathcal{S}', \boldsymbol{K}_3', 8)$, $\mathcal{F}'(\mathcal{S}', \boldsymbol{K}_3', 1)$ and $\mathcal{F}'(\mathcal{S}', \boldsymbol{K}_3', 5)$. One now finds $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 5) = \mathcal{F}'(\mathcal{S}', \boldsymbol{K}_3', 6)$, $\mathcal{F}(\mathcal{S}, \boldsymbol{K}_3, 2) = \mathcal{F}'(\mathcal{S}', \boldsymbol{K}_3', 8)$, etc., which verifies relation (5) for the pair $\{\boldsymbol{K}_3, \boldsymbol{K}_3'\}$. In particular, there are only two admissable labelings for a graph G'. Here again, one can directly verify relation (1). At this point, only two admissable graph labelings should be considered. Alternatively, one can proceed with the remaining pairs $\{\boldsymbol{K}_i, \boldsymbol{K}_i'\}$. In the latter case, one finds that all these pairs satisfy the relation (5), and the sets $\mathcal{S}_\nu$ and $\mathcal{S}_\nu'$ ($\nu = 1, 2$) remain unchanged. This proves that G and G' are isomorphic. The existence of two different sets $\mathcal{S}_\nu$ and two different sets $\mathcal{S}_\nu'$ after the relation (5) is verified for all pairs $\{\boldsymbol{K}_i, \boldsymbol{K}_i'\}$ implies the existence of the nontrivial automorphism group $\mathcal{G}$. We return now to this point.

The main difference in the treatment of a graph automorphism and graph isomorphism problem is that in the former case one has to consider only a single graph and a single adjacency matrix.

For example, in the case of the graph G in fig. 1, initially all 10! labelings are admissable elements of the set $\mathbb{S}_0$. Hence, there are 10! permutations which are potential elements of the automorphism group $\mathcal{G}$. The eigenvector Ψ_1 generates the set $\mathbb{S}_1$ which reduces the number of admissable labelings to 8. The potential elements of a group $\mathcal{G}$ are cyclic permutations (6, 10), (7, 9) and (3, 8), as well as each product of these permutations. The eigenvector Ψ_2 eliminates the permutation (3, 8) as a potential element of a group $\mathcal{G}$, since it discriminates between vertices (3) and (8). The remaining four admissable labelings of the graph G can be directly verified using the relation (1′). Alternatively, one can proceed with the projection operator $\boldsymbol{K}_3$. As shown above, this projection operator generates two sets $\mathcal{S}_\nu$ ($\nu = 1, 2$) which reduces the number of admissable labelings of a graph G to only two. At this point, one can also either proceed with the remaining projection operators $\boldsymbol{K}_i$, or one can directly verify the relation (1′). In either case, one finds that both labelings are admissable, and G hence contains a nontrivial automorphism group $\mathcal{G}$. This group has only two elements, the identity and a permutation $\mathbf{P} = (6, 10)\,(7, 9)$ which is a product of two cyclic permutations (6, 10) and (7, 9).

The above example also illustrates the construction of a unique graph ID. We have considered the projection operators $\boldsymbol{K}_i$ according to the descending sequence of the corresponding eigenvalues λ_i, i.e. in a graph invariant way. The final set $\mathbb{S} \equiv \mathbb{S}_3 \equiv \{\mathcal{S}_1, \mathcal{S}_2\}$ hence determines graph invariant labelings of the graph G. Both labelings define the same adjacency matrix $\mathbf{A}$. One can write the elements of the upper triangle of this matrix sequentially as a series of zeros and ones, and one can

represent each three digits of this sequence with a single octal number. This produces the number 700,024,402,010,410, which is a unique ID of a graph G.

4. Conclusion

A new approach to the problems of graph isomorphism, graph automorphism and the construction of a unique graph ID is suggested. The approach is based on a spectral decomposition $\mathbf{A} = \sum \lambda_i \boldsymbol{K}_i$ of the adjacency matrix **A**.

In order to determine whether graphs G and G' are isomorphic, one first diagonalizes the corresponding adjacency matrices **A** and **A'** and forms spectral decompositions $\mathbf{A} = \sum \lambda_i \boldsymbol{K}_i$ and $\mathbf{A}' = \sum \lambda_i' \boldsymbol{K}_i'$ of these matrices. Next, one compares **A** and **A'** for isospectrality. If these matrices are not isospectral, the algorithm is terminated, and G and G' are found not to be isomorphic. Otherwise, one verifies each pair $\{\boldsymbol{K}_i, \boldsymbol{K}_i'\}$ for the condition (5). This verification involves the comparison of some ordered sequences constructed out of matrix elements of projection operators $\boldsymbol{K}_i$ and $\boldsymbol{K}_i'$. If any of these tests fail, the algorithm is terminated, and G and G' are not isomorphic. In addition, each successful test usually reduces the number of admissable labelings of G and G'. If at any point of the algorithm this number is acceptably small, one can verify isomorphism directly using relation (1). Otherwise, one proceeds with the remaining pairs $\{\boldsymbol{K}_i, \boldsymbol{K}_i'\}$. If all these pairs satisfy (5), graphs G and G' are isomorphic.

Essentially the same algorithm applies to the problem of graph automorphism. The main difference is that instead of two graphs and two adjacency matrices, one has to consider a single graph G and a single adjacency matrix **A**. Each projection operator $\boldsymbol{K}_i$ reduces the number of admissable labelings of G. After all projection operators have been considered, the remaining admissable labelings are equivalent, and they determine the automorphism group $\mathcal{G}$ of G. Here again, one can in many cases terminate the algorithm long before all projection operators are considered. If at any point of the algorithm the number of admissable labelings is acceptably small, one can directly verify potential elements of the automorphism group using relation (1').

A unique graph ID can be obtained along the same lines. Each of the admissable graph labelings can be used in order to generate a unique graph code. Here one has to consider projection operators $\boldsymbol{K}_i$ in the order defined by the descending sequence of the corresponding eigenvalues λ_i. This ensures that the resulting admissable labelings are graph invariants. These labelings define the automorphism group, and hence all these labelings determine the same adjacency matrix **A**. We write the upper triangle of this matrix as a continuous string of zeros and ones. Next we replace each three digits of this string with an octal digit. The resulting string is a unique ID of G.

It is straightforward to generalize this approach to weighted graphs, which might represent molecules with heteroatoms and similar structures. This can be done in basically two different ways. The first method is to incorporate different weights directly in the adjacency matrix **A**. In this case, the algorithms for graph

isomorphism and graph automorphism remain basically the same. The second method is to consider the reference nonweighted graph, and to incorporate different weights after one has obtained canonical labelings of the reference graph. If the reference graph has a unique canonical labeling, this approach is straightforward. If the reference graph has more than one canonical labeling, then one can use graph weights to discriminate between these labelings. This works because the nonweighted graph generates all admissable labelings. These labelings, in conjuncture with different weights, can then be used in order to determine graph isomorphism, graph automorphism, and a unique graph ID.

More generally, the method can be generalized to arbitrary (edge weighted, vertex weighted and oriented) graphs. This might require the distinction between left and right eigenvectors of **A** (if G is oriented), but otherwise the generalization is straightforward.

References

[1] F. Harary, *Graph Theory* (Addison–Wesley, Reading, MA, 1972).
[2] L.C. Ray and R.A. Kirch, Science 126(1957)814.
[3] A.T. Balaban and F. Harary, J. Chem. Doc. 11(1971)258.
[4] H.P. Schultz, E.B. Schultz and T.P. Schultz, J. Chem. Inf. Comput. Sci. 30(1990)27.
[5] D.H. Rouvray, J. Comp. Chem. 8(1987)470.
[6] L. Babai, D. Grigorjev and D.M. Mount, *Proc. 14th Annual ACM Symp. on Theory of Computing*, San Francisco (1982), p. 310.
[7] D. Cvetković, *Linear Algebra and its Application* (Elsevier Science, New York, 1988).
[8] D. Cvetković, *Proc. 8th Yugoslav Seminar on Graph Theory*, Novi Sad (1987).
[9] H. Morgan, J. Chem. Doc. 5(1965)107.
[10] H. Margenau and G. Murphy, *The Mathematics of Physics and Chemistry*, 2nd ed. (Van Nostrand, Princeton, 1956), p. 502.
[11] M. Randić, J. Chem. Inf. Comput. Sci. 15(1976)105; 17(1977)171.
[12] M. Razinger, Theor. Chim. Acta 61(1982)581.
[13] W.C. Herndon and J.E. Leonard, Inorg. Chem. 22(1983)554.
[14] W.C. Herndon and S.H. Bertz, J. Comp. Chem. 8(1987)367.
[15] D. Weininger, A. Weininger and J.L. Weininger, J. Chem. Inf. Comput. Sci. 29(1989)97.
[16] F.R. Burden, J. Chem. Inf. Comput. Sci. 29(1989)225.
[17] M. Randić, J. Chem. Inf. Comput. Sci. 26(1986)134.
[18] M. Randić, S. Nikolić and N. Trinajstić, J. Mol. Struct. (THEOCHEM) 165(1988)213.
[19] M.I. Stankevitch, S.S. Tratch and N.S. Zefirov, J. Comp. Chem. 9(1988)303.
[20] N.S. Chaudhari and D.B. Phatak, Jr., Comb. Inf. System Sci. 13(1988)1.
[21] D.G. Corneil and C.C. Gotlieb, J. ACM 17(1970)51.
[22] C. Jochum and J. Gasteiger, J. Chem. Inf. Comp. Sci. 17(1977)113.
[23] J.V. Knop, W.R. Müller, K. Szymanski, K.W. Kroto and N. Trinajstić, J. Comp. Chem. 8(1987)549.
[24] W.R. Müller, K. Szymanski, J.V. Knop, S. Nikolić and N. Trinajstić, J. Comp. Chem. 11(1990)223.
[25] T.P. Živković, in preparation.
[26] T. Kato, *Perturbation Theory for Linear Operators* (Springer, Berlin, 1966).
[27] T.P. Živković, N. Trinajstić and M. Randić, Mol. Phys. 30(1975)517;
M. Randić, N. Trinajstić and T.P. Živković, J. Chem. Soc., Faraday Trans. 72(1976)244;
W.C. Herndon and M.L. Ellzey, Jr., Tetrahedron 31(1975)99, and references in these papers.
[28] D. Klein, private communication.