

Graph Automorphism Perception Algorithms in Computer-Enhanced Structure Elucidation

Marko Razinger,[†] K. Balasubramanian,[‡] and Morton E. Munk*

Department of Chemistry, Arizona State University, Tempe, Arizona 85287-1604

Received March 2, 1992

The concept of graph symmetry is explained in terms of the vertex automorphism group, which is a subgroup of the complete vertex permutation group. The automorphism group can be deduced from the automorphism partition of graph vertices. An algorithm is described which constructs the automorphism group of a graph from the automorphism vertex partitioning. The algorithm is useful especially for graphs which contain more than one vertex-partition set. Several well-known topological symmetry perception algorithms that yield automorphism partitions are compared. The comparison is favorable to the Shelley-Munk algorithm, developed in the framework of the SESAMI system for computer-enhanced structure elucidation.

INTRODUCTION

Most computer programs that require the manipulation of chemical structures among their operations (e.g., computer-enhanced structure elucidation, spectrum interpretation and simulation, and synthetic design) must eventually be able to recognize and process stereochemical information if they are to be truly useful.¹ There are different approaches to the treatment of stereochemistry² and related problems, but all of them share in common the need to consider the symmetry of the molecular structure under consideration.

Symmetry properties of molecular structures are very important because they are reflected in a variety of molecular properties, interactions, and other structure-related phenomena.³ For the correct interpretation of some experimentally derived information, e.g., X-ray diffraction patterns or IR spectra, the determination of molecular stereochemistry is necessary; this implies the use of group theory formalism in the determination of appropriate space or point groups and their elements. In some other areas, like computer-enhanced structure elucidation⁴ or the enumeration of ¹³C-NMR signals,⁵ partial solutions are possible using topological symmetry (connectivity) instead of three-dimensional geometry. For such cases, a different concept of symmetry, based on graph theory, can be used to advantage.⁶

The objective of this article is to describe the connection between the automorphism partitioning of the vertices of graphs and the automorphism groups of graphs. An algorithm to construct the automorphism group from the automorphism partition of the vertices is presented. Several procedures to obtain the automorphism partitioning of graphs, including graphs containing isospectral points, are compared with the Shelley-Munk algorithm.^{7,8} A critical comparison shows, contrary to the perception in the literature,^{9,10} that the algorithm is rigorous and efficient and distinguishes isospectral points without difficulty.

SYMMETRY OF GRAPHS

The properties of a graph do not depend on its pictorial representation, but only on its connectivity. Thus, a graph, which can be drawn in several different ways, may appear to possess different symmetry elements in the normal sense of point group symmetry; however, the topological symmetry of each representation is identical based on connectivity. This

aspect makes the consideration of graph symmetry quite different from the conventional molecular point group and space group approaches. (This is discussed with examples in ref 11.)

In terms of graph theory, the structural symmetry relates to the vertex automorphism group: this group is a subgroup of the complete vertex permutation group of the graph representing the chemical structure.^{12,13} Mathematically, these relations can be expressed simply and succinctly by using matrix algebra. The two matrices used in the following relations are the adjacency matrix *A* and the permutation matrix *P* defined as

$$A_{ij} = \begin{cases} 1 & \text{if vertices } i \text{ and } j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

$$P_{ij} = \begin{cases} 1 & \text{if a mapping of vertices } i \text{ --- } j \text{ is induced by the given} \\ & \text{permutation of vertices} \\ 0 & \text{otherwise} \end{cases}$$

There are *n*! possible labelings of a graph with *n* vertices. Each of them can be represented by a vector *P*, and the transformation between two different labelings is given by the permutation matrix *P*.

In Figure 1, three of 120 possible labelings of a graph are shown together with the corresponding adjacency matrices *A* and permutation vectors *P*. Although the graphs *G*₁ and *G*₂ represent the same structure, they are not identical because of the different labelings of their vertices, i.e., different connectivity. Since they represent the same structure, they are known as isomorphic graphs. The isomorphism can be expressed mathematically with the following relations, holding generally for all graphs:

$$P_{12}^T \cdot A(G_1) \cdot P_{12} = A(G_2) \quad (1)$$

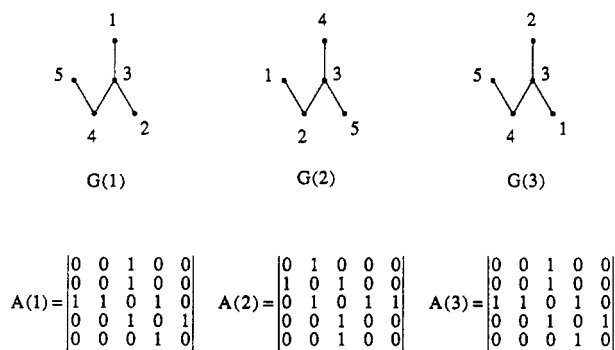
Because the permutation matrix is an orthogonal matrix, its inverse is equal to its transpose, and the relation can be further simplified:

$$P_{12}^T \cdot A(G_1) \cdot P_{12} = A(G_2) \quad (2)$$

Such a transformation is called an isomorphism between the graphs *G*₁ and *G*₂. Figure 2 shows the matrices and the results of matrix multiplication for the expression 2: the permutation matrix *P*₁₂ expresses the mapping *P*₁ - - *P*₂ and the consecutive multiplication of *A*(*G*₁) with *P*₁₂ and *P*₁₂^T results in *A*(*G*₂).

[†] Present address: Institute of Chemistry, 61115 Ljubljana, Hajdrihova 19, Slovenia.

[‡] Camille and Henry Dreyfus Teacher-Scholar.



$$P(1) = (1 \ 2 \ 3 \ 4 \ 5) \quad P(2) = (4 \ 5 \ 3 \ 2 \ 1) \quad P(3) = (2 \ 1 \ 3 \ 4 \ 5)$$

Figure 1. Illustration of graph isomorphism and automorphism.

$$P(12) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P^{-1}(12) = P^T(12) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A(1) \cdot P(12) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$P^T(12) \cdot A(1) \cdot P(12) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = A(2)$$

Figure 2. Result of multiplying $P^T(12)$, A , and $P(12)$.

If there exists a transformation P which transforms the graph G into itself (an isomorphism of a graph with itself), such a transformation G is called an automorphism, denoted in the matrix relation as:

$$P^T \cdot A(G) \cdot P = A(G) \quad (3)$$

Every graph possesses at least one automorphism, i.e., the one induced by the identity permutation. Besides the identity transformation, the graph G_1 in Figure 1 has only one automorphism which corresponds to the graph G_3 . The fact that the two graphs G_1 and G_3 are identical is shown by the identity of A_1 and A_3 ; it can be verified also by the application of eq 3 using the permutation P_3 .

Since every relabeling of the graph vertices corresponds to a vertex permutation, every symmetry operation on a graph can be represented by an automorphism, and inversely, every automorphism corresponds to some symmetry transformation of the same graph. Hence, it follows that the overall symmetry of any given graph (often called topological symmetry in the chemical literature) can be recognized by the identification of all graph automorphisms which together form the automorphism group.

AUTOMORPHISM GROUP AND AUTOMORPHISM PARTITION

For some applications, like approximating the number of ^{13}C -NMR signals expected for a compound of given structure, it is not necessary to identify the automorphism group of the graph. Instead, it is sufficient to identify the automorphism partition of the vertices of the graph. This partition separates the vertices into equivalence classes, which in many cases correspond to what are referred to as magnetic equivalences in the NMR literature. The vertices belonging to the same

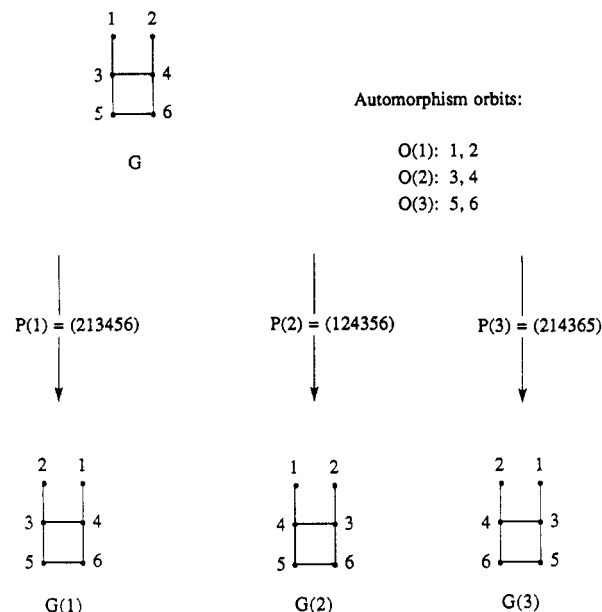


Figure 3. Illustration of the generation of automorphism partition of the vertices.

equivalence class cannot be distinguished by their graph-theoretical parameters. This property of automorphism partition is known also as the topological symmetry.¹⁴

There is an important difference between the information content of the automorphism partition and the automorphism group which must be always kept in mind. All vertex permutations implicit in the automorphism partition (for example, the pairwise interchange of two vertices belonging to the same class) lead to isomorphic or automorphic graphs. Only those permutations which preserve the connectivity of the graph (or equivalently do not make or break an edge in the graph) as defined by eq 3 belong to the automorphism group.

This is illustrated in Figure 3, showing a graph G and its automorphism partition. There are three classes in the partition. Three permutations of vertices within the same classes are shown (out of the total number of 6! different labelings and 8 possible combinations of intraorbit permutations). While permutations P_1 and P_2 lead to isomorphism, only P_3 leads to automorphism since graph G_3 is identical to the starting graph G . From the comparison of G and G_3 , one can see also that the automorphism P_3 represents a symmetry operation. That is, the permutation matrix which corresponds to P_3 has satisfied eq 3.

Permutation P_3 is also the only automorphism of the graph G in Figure 3 besides the identity operation. The comparison of G_1 , G_2 , and G_3 offers another more pragmatic definition of automorphism as a permutation of graph vertices which does not disconnect any edge in the graph. This definition, while being more "graphic", is equivalent to the statement that the adjacency matrix of a graph does not change after an automorphic permutation of vertices.

ALGORITHMS FOR AUTOMORPHISM PARTITIONING

The Morgan algorithm¹⁵ is perhaps the first, and certainly the most known, of the vertex partitioning algorithms. Although originally developed to uniquely number the atoms of a chemical structure and not for automorphism partitioning, the algorithm has been used for the latter and is still popular because of its simplicity. The algorithm makes use of the concept of extended connectivity, the graph-theoretical basis

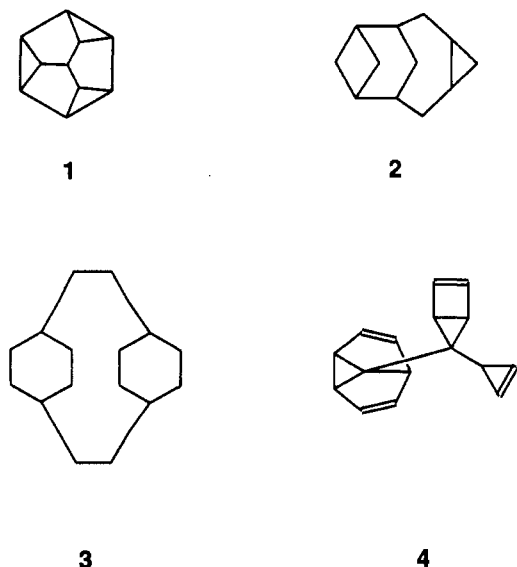


Figure 4. Some complex graphs.

of which was described later.¹⁶ A number of modifications of the original algorithm by other authors has improved its performance and applicability.^{7,8,17,18}

The latest procedure of Shelley and Munk,⁸ a modification and extension of their earlier algorithm,⁷ is a combination of vertex classification (by the modified Morgan algorithm), subsequent depth-first construction of sequence number permutations, and the final use of automorphism to restrict the construction process. The automorphism partitioning is carried to its logical conclusion by calculating the necessary vertex permutations. Although one could label such a method as a "brute force" method, it is far from that indeed; the combination of vertex classification and the early detection of graph symmetry by the depth-first search method of graph labeling reduces the number of permutations drastically and makes the procedure efficient.⁸

The modified Shelley-Munk algorithm⁸ has now been tested with a wide range of "difficult" graphs starting with those which were given as counterexamples (Figure 4) for the original algorithm in a paper by Carhart.¹⁹ The modified algorithm, published 2 years later in 1979, does determine the automorphism partition correctly for these counterexamples. The same holds true for all other examples described below (Figures 5-8).

A review of the literature on topological symmetry perception revealed several papers which appeared years after the modified Shelley-Munk⁸ was published that did not cite it. Instead, referring to the original paper,⁷ they report the failure of the algorithm in the case of certain graphs, when in fact the modified algorithm performs correctly. We believe that such statements discourage use of a fast, rigorous algorithm that can be used advantageously in many chemical applications. Herndon⁹ in 1983, referring to the earlier algorithm⁷ states: "A difficulty that remains in both methods [the second being the Morgan algorithm¹⁵] is that isospectral points will not be distinguished by either procedure". As an example, he cites graph (a) of Figure 5, with the isospectral points highlighted. As shown in the work described here, the statement is incorrect for the modified Shelley-Munk algorithm. Balaban et al.¹⁰ in 1985, also failing to recognize the 1979 paper, reinforced the perception of a flawed algorithm, incorrectly citing graph (b) (Figure 5) as a counterexample.

Balaban et al.¹⁰ compared the performance of three algorithms for vertex discrimination—ECA [Extended Connectivity Algorithm (Morgan Algorithm)], SEMA¹⁷ (Ste-

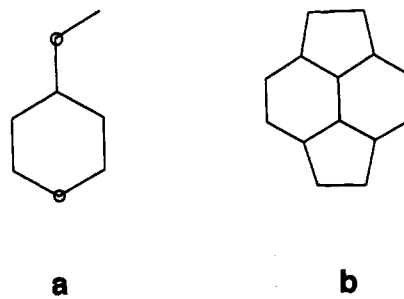


Figure 5. Graphs containing isospectral points.

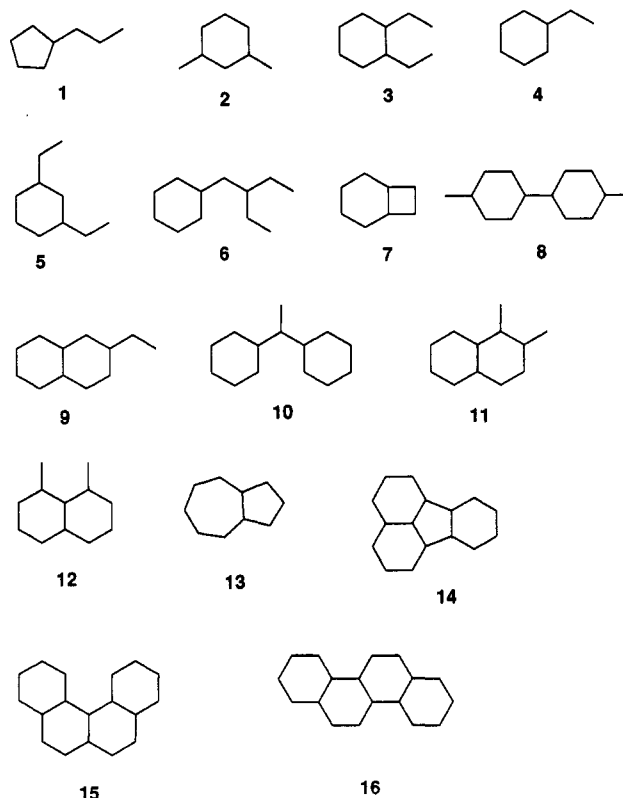


Figure 6. Test graphs.

Table I. Comparison of Performance of Various Algorithms for Automorphism Vertex Partitioning of Graphs in Figure 6

structures	ECA	SEMA	HOC-1	S-M
1	5	5	6	6
2	4	4	5	5
3	4	5	5	5
4	5	6	6	6
5	5	6	6	6
6	7	8	8	8
7	4	4	4	4
8	5	5	5	5
9	9	9	12	12
10	6	6	6	6
11	11	10	12	12
12	6	6	7	7
13	6	6	6	6
14	9	9	9	9
15	10	10	10	10
16	9	9	9	9
sum	105	108	116	116

reochemically Extended Morgan Algorithm), and HOC (Hierarchically Ordered Extended Connectivities)—using the 16 graphs shown in Figure 6. Algorithms were compared by the number of vertex equivalence classes predicted for each of the graphs (Table I). The results obtained for those graphs using the Shelley-Munk algorithm⁸ have been added to those of Balaban. The Shelley-Munk and HOC algorithms perceived all automorphism equivalence classes for all 16 graphs,

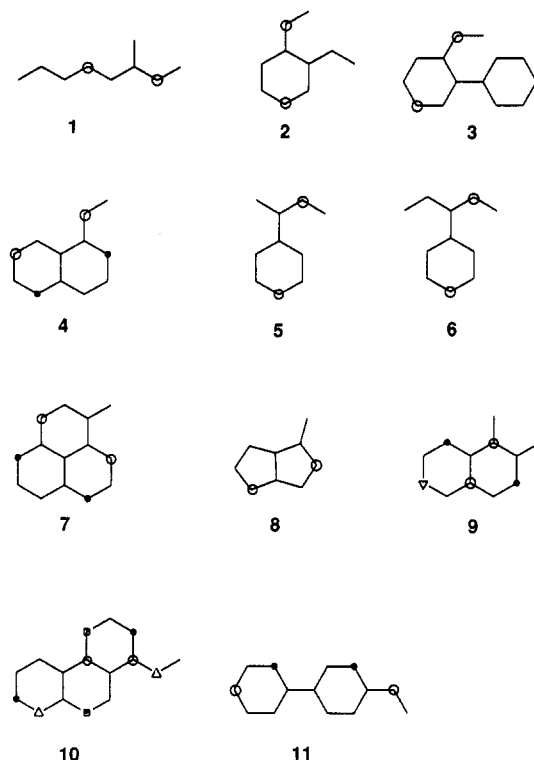


Figure 7. Graphs containing isospectral points.

while the ECA and SEMA algorithms failed in nine and five of the examples, respectively. The sum of the equivalence classes for all 16 graphs (last row, Table I) can be viewed as a comparative measure of overall performance of these algorithms.

Finally, the Shelley-Munk algorithm was tested against a series of graphs containing isospectral points and highly regular graphs. The sets of graphs were taken from papers by Liu et al.²⁰ and by King,²¹ respectively. It is well-known that the Morgan algorithm (and related procedures) has problems with graphs containing isospectral points since the extended connectivity values of isospectral points are equal. The modified Shelley-Munk algorithm yields complete separation of all isospectral points in the graphs shown in Figure 7.

Regular graphs are graphs in which every vertex has the same degree. Figure 8 shows 19 regular graphs of degree 3. The correct number of automorphism equivalence classes was obtained by the Shelley-Munk algorithm for each and every one of those graphs.

Table II shows the actual CPU times in seconds for some graphs in Figures 6–8 obtained by implementing the Shelley-Munk algorithm on a VAX workstation 3100. As seen from Table II for all graphs in Figure 6, the CPU times range from 0.01 to 0.09 s. The existence of isospectral points did not introduce any additional complications or increase in CPU times as seen from the CPU times for graphs in Figure 7 which contain multiple isospectral points. The most complex of the graphs among the ones in Figure 8 are those which contain several fused (clustered) cycles such as graphs labeled 12, 13, 15, 17, 18, and 19. The CPU time for even these graphs were 0.12–0.28 s; the maximum CPU time being for the graph number 17 (0.28 s). As seen from Figure 8, the graph labeled 17 is quite clustered. On the basis of the CPU times and the correctness of the codes for these cases, we conclude that the Shelley-Munk algorithm performs as well as other state-of-the-art algorithms for graph automorphism perception.

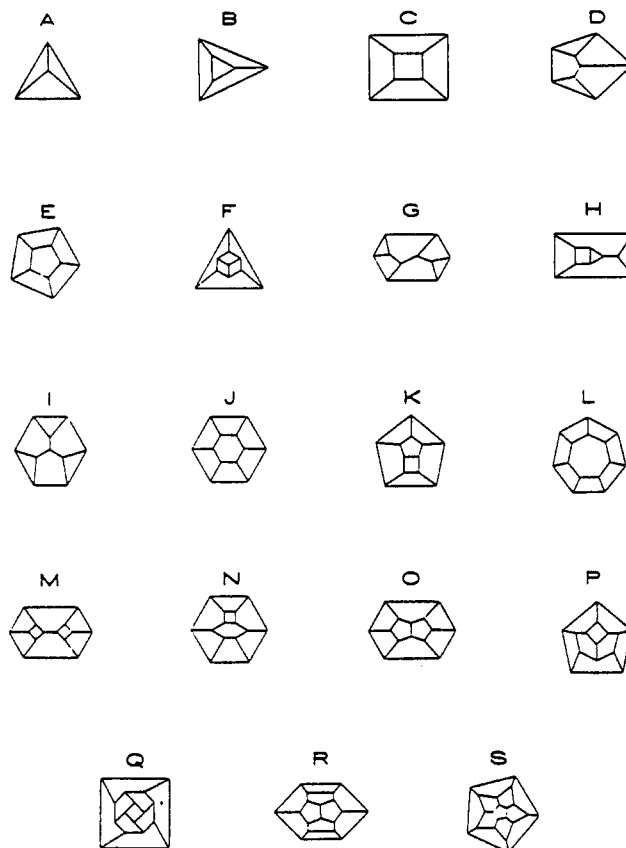


Figure 8. Some regular graphs.

Table II. CPU Times for Automorphism Partitioning Using Shelly-Munk Algorithm for Some Graphs in Figures 6–8^a

graph	CPU (s)	graph	CPU (s)
Figure 6 (1)	0.01	Figure 7 (2)	0.03
Figure 6 (2)	0.02	Figure 7 (3)	0.03
Figure 6 (3)	0.03	Figure 7 (4)	0.03
Figure 6 (4)	0.02	Figure 7 (5)	0.03
Figure 6 (5)	0.04	Figure 7 (6)	0.05
Figure 6 (6)	0.04	Figure 7 (7)	0.02
Figure 6 (7)	0.02	Figure 7 (8)	0.04
Figure 6 (8)	0.04	Figure 7 (9)	0.03
Figure 6 (9)	0.02	Figure 7 (10)	0.01
Figure 6 (10)	0.09	Figure 7 (11)	0.05
Figure 6 (11)	0.02	Figure 8 (N)	0.12
Figure 6 (12)	0.08	Figure 8 (O)	0.13
Figure 6 (13)	0.06	Figure 8 (P)	0.13
Figure 6 (14)	0.05	Figure 8 (Q)	0.28
Figure 6 (15)	0.06	Figure 8 (R)	0.17
Figure 6 (16)	0.06	Figure 8 (S)	0.12
Figure 7 (1)	0.03		

^a All CPU times are in seconds on a VAX workstation. The number in parenthesis is the structure number in the corresponding figure.

ALGORITHMS FOR CONSTRUCTION OF AUTOMORPHISM GROUP FROM AUTOMORPHISM PARTITION

Most of the algorithms developed to date yield only the automorphism partitioning of the vertices of a graph. As indicated earlier, the automorphism group and automorphism partitioning of vertices are not the same. While knowledge of the automorphism group can provide the automorphism partitioning of the vertices, the converse is not true. All algorithms described up to now, however, yield only the automorphism partitioning of the vertices of a graph. The question is how this information can be related to the automorphism group. An algorithm has been designed to generate the automorphism group of a graph from the automorphism partitioning of the vertices. This algorithm

reduces the $n!$ checking (n = number of vertices) to a much smaller number for several graphs of chemical interest.

Suppose the vertices of a graph G are partitioned into sets V_1, V_2, \dots, V_m under the automorphism. Let the number of elements in V_i , denoted by $|V_i|$, be v_i and the total number of vertices in the graph be n . Then an automorphism permutation can induce a permutation on the vertices of G such that all its orbits are within the same V_i set. That is, a permutation P of the vertices of a graph cannot be an automorphism permutation if it interchanges an element in a set V_i with another element in a set V_j ($i \neq j$). The automorphism group of any graph is a subgroup of S_n where n is the number of vertices and S_n is the set of all $n!$ permutations of n objects. However, if the vertices of G are partitioned into more than one V set as above through the Shelley–Munk algorithm or other algorithms, then the automorphism group of G , denoted by Γ , has to be a subgroup of $S_{v_1} \times S_{v_2} \times \dots \times S_{v_m}$, where $v_i = |V_i|$. Symbolically

$$\Gamma \subseteq S_{v_1} \times S_{v_2} \times \dots \times S_{v_m}$$

Our algorithm generates permutations in the group $S_{v_1} \times S_{v_2} \times \dots \times S_{v_m}$ and checks if a $P \in S_{v_1} \times \dots \times S_{v_m}$ satisfies

$$P^T \cdot A \cdot P = A$$

where A is the adjacency matrix of the graph under consideration. The maximum number of such checkings can be seen to be $v_1! \times v_2! \times \dots \times v_m!$. However, in practice one can use the properties of a group to eliminate numerous checkings. Suppose P_1, P_2, \dots, P_k already satisfies the automorphism condition. Then it is clear that $P_1^{-1}, P_2^{-1}, \dots, P_k^{-1}$ must belong to Γ since Γ is the automorphism group. Likewise, all possible products $P_i P_j$ ($i = 1 - k; j = 1 - k$) must also belong to Γ . It is also true that $P_i^2, P_i^3, \dots, P_i^m \neq e$ (identity) must be in Γ . This property can be used if a P_i happens to contain a longer orbit. Implementation of these procedures in the algorithms to construct Γ can dramatically decrease the computational time in generating Γ . Of course, in the worst case when the graph contains only one automorphism partitioned vertex set, the number of computations in the algorithm can be large. However, while such graphs occur in isomerizations of chemical reactions (reaction graphs), they are not commonly found in structure elucidation problems.

A version of this algorithm was implemented on a VAX workstation. Although the present code does not exploit the closeness and the existence of inverse elements in a group, we found that the algorithm does yield the correct automorphism group for all the graphs considered in this study as well as other graphs of potential use in stereoisomer generation. The CPU times for generating the automorphism groups of most of the graphs in Figure 6 were less than 0.5 s on a VAX workstation 3100. The CPU times for graphs 8 and 14–16 in Figure 6 were somewhat larger (4.5–90 s), the largest being for graph 8 in Figure 6. The CPU times for graphs in Figure 4 for generating the automorphism groups were between 5 and 12 s on a VAX workstation 3100. The first two graphs in Figure 8 took 0.3 and 0.8 s of CPU time, respectively. The graph 9 took 7.8 s while the graph 13 took 2.8 s. Thus, we conclude that the CPU times are small for graphs with low symmetry and somewhat larger for graphs with higher symmetry. The automorphism groups thus generated are being used to enumerate and construct the stereoisomers of organic compounds. This topic will be discussed further in a future publication.

CONCLUSION

The Shelley–Munk algorithm for the perception of topological symmetry was published first in 1977 and in a modified and improved version in 1979. It is a fast, efficient, and rigorous procedure for automorphism partitioning of graph vertices. It has been shown in this paper that the modified algorithm for topological symmetry perception of Shelley and Munk⁸ can correctly solve the automorphism partitioning of all pathological graphs, including graphs containing isospectral points found in later papers.^{9,10,20–22} Additionally, an algorithm to generate the automorphism group of a graph from its automorphism partition of the vertices has been described.

ACKNOWLEDGMENT

The financial support of this work by the National Institutes of Health (GM 37963) and Sterling Drug, Inc. is gratefully acknowledged.

REFERENCES AND NOTES

- (1) (a) Shelley, C. A.; Woodruff, H. B.; Snelling, C. R.; Munk, M. E. In *Computer-Assisted Structure Elucidation*; Smith, D. H., Ed.; American Chemical Society: Washington, DC, 1977; Vol. 54, Chapter 7, pp 92–107. (b) Gray, N. A. B. *Computer-Assisted Structure Elucidation*; John Wiley: New York, 1986; Chapters 7 and 10.
- (2) (a) Mislow, K. *Introduction to Stereochemistry*; W. A. Benjamin: New York, 1966. (b) Cahn, R. S.; Ingold, C.; Prelog, V. *Angew. Chem., Int. Ed. Engl.* **1966**, *5*, 385.
- (3) Balasubramanian, K. Applications of Combinatorics and Graph Theory to Spectroscopy and Quantum Chemistry. *Chem. Rev.* **1985**, *85*, 599–618.
- (4) Christie, B. D.; Munk, M. E. The Role of Two-Dimensional Nuclear Magnetic Resonance in Computer-Enhanced Structure Elucidation. *J. Am. Chem. Soc.* **1991**, *113*, 3750–3757.
- (5) Shelley, C. A.; Munk, M. E. Signal Number Prediction in Carbon-13 Nuclear Magnetic Resonance Spectrometry. *Anal. Chem.* **1978**, *50*, 1522–1527.
- (6) Davis, M. I.; Ellzey, M. L., Jr. A Technique for Determining the Symmetry Properties of Molecular Graphs. *J. Comput. Chem.* **1983**, *4*, 267–275.
- (7) Shelley, C. A.; Munk, M. E. Computer Perception of Topological Symmetry. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 110–113.
- (8) Shelley, C. A.; Munk, M. E. An Approach to the Assignment of Canonical Tables and Topological Symmetry Perception. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 247–250.
- (9) Herndon, W. C. Canonical Labelling and Linear Notation for Chemical Graphs. In *Chemical Applications of Topology and Graph Theory*; King, R. B., Ed.; Studies in Physical and Theoretical Chemistry; Elsevier: Amsterdam, 1983; Vol. 28, pp 231–242.
- (10) Balaban, A. T.; Mekenyan, O.; Bonchev, D. Unique Description of Chemical Structures Based on Hierarchically Ordered Extended Connectivities (HOC Procedures). I. Algorithms for Finding Graph Orbits and Canonical Numbering of Atoms. *J. Comput. Chem.* **1985**, *6*, 538–551.
- (11) Balasubramanian, K. Symmetry Groups of Chemical Graphs. *Int. J. Quantum Chem.* **1982**, *21*, 411–418.
- (12) Harary, F. *Graph Theory*; Addison-Wesley: Reading, MA, 1971.
- (13) Trinajstić, N. *Chemical Graph Theory*; CRC Press: Boca Raton, FL, 1983; Vol. 1, Chapter 4.
- (14) Read, R. C.; Cornell, D. G. The Graph Isomorphism Disease. *J. Graph Theory* **1977**, *1*, 339–363.
- (15) Morgan, H. L. Generation of Unique Machine Description for Chemical Structures a Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107–112.
- (16) Razinger, M. Extended Connectivity in Chemical Graphs. *Theor. Chim. Acta* **1982**, *61*, 581–586.
- (17) Wipke, W. T.; Dyott, T. M. Stereochemically Unique Naming Algorithm. *J. Am. Chem. Soc.* **1974**, *96*, 4834–4842.
- (18) Moreau, G. A Topological Code for Molecular Structures. A Modified Morgan Algorithm. *Nouv. J. Chim.* **1980**, *4*, 17–22.
- (19) Carhart, R. E. Erroneous Claims Concerning the Perception of Topological Symmetry. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 108–110.
- (20) Liu, X.; Balasubramanian, K.; Munk, M. E. Computer-Assisted Graph Theoretical Construction of ¹³C NMR Signal and Intensity Patterns. *J. Magn. Reson.* **1990**, *87*, 457–476.
- (21) King, R. B. The Bonding Topology of Polyhedral Molecules. In *Chemical Applications of Topology and Graph Theory*; King, R. B., Ed.; Studies in Physical and Theoretical Chemistry; Elsevier: Amsterdam, 1983; Vol. 28, pp 108–122.