Computer-Assisted Graph-Theoretical Construction of $^{13}$C NMR Signal and Intensity Patterns

XIAOYU LIU, K. BALASUBRAMANIAN,* AND M. E. MUNK

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

Received May 3, 1989; revised June 28, 1989

An efficient algorithm and computer code are developed to generate the automorphism partitioning of nuclei set $V$ of a given molecular graph. Various graphs of chemical interests are considered, including isospectral graphs. The developed computer code is used to generate $^{13}$C NMR signals and intensity ratios of different organic compounds simply from the neighborhood table of the compounds. The code developed would thus be valuable as an integral part of artificial intelligence methods for computer-assisted structure elucidation. © 1990 Academic Press, Inc.

Chemical applications of graph-theoretical techniques have been the topic of many investigations in recent years (1–10). One of the present authors (10) has reviewed recently applications of combinatorics and graph theory to spectroscopy and quantum chemistry. In the area of NMR, combinatorial techniques have been used for deriving the number of NMR signals, for characterizing the symmetry of the NMR spin Hamiltonian through NMR graphs, and in constructing symmetry-adapted NMR spin functions (1, 10). Randić and co-workers (11, 12) have shown that graph-theoretical techniques could be used to obtain the chemical shifts of nuclei. Grant and Paul (13) have used the traditional bond additivity methods to obtain chemical shifts. Randić and co-workers (14) have also developed a computer code for listing equivalence classes of graphs which works for most of the small, nontransitive, and nonsospectra graphs.

In earlier investigations, Balasubramanian (1, 2) has shown that the number of NMR signals of rigid as well as nonrigid molecules could be obtained from the symmetry of the molecule. One can construct a group-theoretical structure known as the cycle index of a group which enumerates the number of NMR signals. On the basis of this technique Balasubramanian (3) developed a computer code to count the number of NMR signals. Another computer code was later developed also by Balasubramanian to generate the number of NMR signal and intensity patterns from the generators of the symmetry group of the molecule (15). Although many graph vertex partitioning algorithms as well as graph automorphism partitioning algorithms (16–29) have been proposed to investigate some basic chemical graph-theoretical problems such as the canonical labeling problem, the graph isomorphism problem, there are seemingly very few applications of the automorphism partitioning techniques to the

* Camille and Henry Dreyfus Teacher-Scholar.

457
problem of generating NMR signal and intensity patterns. Shelley and Munk (26) have developed a program to predict the number of $^{13}$C NMR signals. In this investigation, we develop a computer code based on powerful automorphism partitioning algorithms to generate NMR signal and intensity patterns. Our ultimate objective in this area is to combine this information, namely the NMR signal and intensity patterns, with a knowledge base or a machine-learning system which would generate simulated NMR spectra given the neighborhood table of the molecule.

Although there are algorithms available at present to generate the automorphism partitioning and to count the $^{13}$C NMR signals, often these algorithms provide special problems for isospectral graphs. Furthermore, many of the existing algorithms also exhibit oscillatory behavior especially for graphs containing a large number of vertices which needs to be circumvented using special procedures. The readers are referred to the article by Herndon (32) for a comparison of existing algorithms.

The objective of this investigation is to develop algorithms and codes that are especially stable and applicable even for isospectral graphs and graphs containing large numbers of vertices. The algorithms and codes developed here are shown to be applicable to enumerate and construct NMR signal patterns of complex organic compounds simply from the neighborhood table. Munk and co-workers (27) have strategies and codes for computer-assisted structure elucidation (Program CASE). Thus, the code developed in the present investigation would become an integral part of the CASE package and enhance the capabilities of this package as regards spectral analysis and simulation. The codes developed here, however, do not take into account the stereochemistry of molecules. Nevertheless, $^{13}$C NMR signal patterns do not depend on stereochemistry as much as proton NMR and, consequently, these codes should be applicable to $^{13}$C NMR of most molecules.

In the following section, we outline the basic algorithms and preliminaries for efficient automorphism partitioning of graphs consisting of large numbers of vertices. The last section comprises results and discussions and applications to the computer-assisted construction of $^{13}$C NMR signal and intensity patterns.

**ALGORITHMS AND PRELIMINARIES**

Given a chemical graph $G(V, E)$, with $V$ denoting the vertex (nucleus) set and $E$ the edge (bond) set, its adjacency matrix is defined as

$$A = (a_{ij}), \quad i, j = 1, 2, \ldots, |V| = n = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise.} \end{cases}$$ \[1\]

An automorphism of a graph $G$ is an isomorphism of $G$ with itself. Thus each automorphism of $G$ is a permutation of the vertex set $V$, which preserves the adjacency. In other words, if $P$ is an $n \times n$ permutation matrix which corresponds to a permutation of the vertices of the graph, then $P$ belongs to the automorphism group if

$$P^{-1}AP = A.$$ \[2\]

The set of all automorphisms of a graph $G$ under the binary operation of composition is always a group, called the automorphism group. The nucleus set $V$ can be parti-
tioned into disjoint equivalence classes by the automorphism group of the graph under study, namely

$$V = \bigcup_{i=1}^{k} V_i, \quad V_i \cap V_j = \emptyset \quad \text{for} \quad i \neq j,$$

where $k$ is the number of equivalence classes; this number is also the number of NMR signals considering that two nuclei in the same equivalence class are magnetically equivalent and thus would resonate at the same magnetic field. Since all the elements in the same class resonate at the same applied field, the intensity ratio of the various signals is simply the ratio of the number of elements in classes $V_1, V_2, \ldots, V_k$. Thus, the number of equivalence classes under the automorphism and the number of elements in each equivalence class would provide for the number of NMR signals and NMR intensities. In this investigation the NMR signal and intensity patterns are generated by a computer code based on two nonnumerical algorithms which we are going to discuss very briefly in the ensuing paragraphs.

**Algorithm I**

The vertices in $V$ are first classified into subsets by degree partitioning, namely, $V = V_1 \cup V_2 \cup \ldots \cup V_k$, where vertices $i$ and $j$ belong to the same subset if and only if their degrees are the same. Obviously if nucleus set $V$ contains only carbon atoms, then the maximum number of subsets the degree partitioning could achieve is $k = 4$.

Now to each vertex $i$ assign a list $L_i = (a_1, a_2, \ldots, a_k)$, where $a_j (1 \leq j \leq k)$ is the number of vertices in the subset $V_j$ adjacent to $i$, or more precisely $a_j$ equals the exact number of nuclei $k \in V_j$ such that $(i, k) \in E$. Using these lists, each $V_j$ may now be partitioned into subsets, each subset consisting of all vertices with a given list. In this way, we refine the original collection of subsets. No refinement will be obtained if all vertices in each subset have identical lists. If there is a change in the partitioning of $V$, the method is iterated until there is no further change in the partitioning of $V$.

Below we give a full description of this iterative vertex classification algorithm in the form of a pseudocode.

**Module name:** Vertex_Partion

**Function:** Generating an initial partitioning among nuclei in $V$

Get $n$, and adjacency matrix $A = (a_{ij}), i, j = 1, \ldots, n$

Set $V_i = \emptyset, 1 \leq i \leq n$! $V_i$'s are subsets under vertex partitioning

**Step 1. Degree partitioning**

$d_{\text{max}} = 0$! $d_{\text{max}}$ is the largest vertex degree

For $i = 1$ to $n$ do

\[ d = 0 \]

For $j = 1$ to $n$ do

\[ \text{If } a_{ij} \neq 0 \text{ then } d = d + 1 \]

End do

\[ V_d = V_d \cup \{i\} \]

\[ \text{If } (d > d_{\text{max}}) d_{\text{max}} = d \]

End do

\[ t' = d_{\text{max}} \]
Step 2. List generation
\[ L_i(j) = 0, \; 1 \leq i \leq n, \; 1 \leq j \leq t' \] is vertex i's list
For \( i = 1 \) to \( n \) do
  For all \( k \in V_i \) if \((k, i) \in E\) then
    For \( j = 1 \) to \( t' \) do
      If \( k \in V_j \) then \( L_i(j) = L_i(j) + 1 \)
    End do
  End if
End do

Step 3. Subset refinement
\( t = 0 \)
For \( i = 1 \) to \( t' \) do
  If \( V_i = \emptyset \) then go to 501
  If \( |V_i| = 1 \) then
    \( t = t + 1; E_i = \emptyset; E_{i^*} = E_i \cup V_i; V_i = \emptyset \) !E's are new subsets of \( V \)
  End if
  Get \( p \in V_i; t = t + 1; E_i = \emptyset; E_{i^*} = E_i \cup \{ p \}; V_i = V_i - \{ p \} \)
    For all \( q \in V_i \) do
      If \( L_p = L_q \) then
        \( E_i = E_i \cup \{ q \}; V_i = V_i - \{ q \} \)
      End if
    End do
  If \( V_i \neq \emptyset \) then go to 501
End do
If \( t \neq t' \) then
  \( t' = t \)
  For \( i = 1 \) to \( t' \) do
    \( V_i = E_i \)
  End do
Go to Step 2
End if
End module

The two iterative steps, i.e., Step 2 and Step 3, are repeated until no subset is refined, that is, until the condition \( t = t' \) is satisfied, and the initial vertex partitioning of nucleus set \( V \) with respect to the degree partitioning has been obtained. Under the degree partitioning, two vertices belong to the same subset if and only if their degrees are the same. Therefore if the given chemical graph \( G \) is regular, the degree partitioning consists of the vertex set \( V \) alone, and then the two-step iteration does not refine the partitioning; consequently no classification of nucleus set \( V \) is achieved. Below we demonstrate the algorithm with the graph 1 in Fig. 1 with its isospectral points circled. By the way, in this paper, all the graphs with sequence number assignments are randomly labeled.
Fig. 1. Two examples of graphs, one of which is isospectral (circled vertices are isospectral points).

Example: graph 1 in Fig. 1
From Step 1:

<table>
<thead>
<tr>
<th>Subset index</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>2, 4, 5, 6, 7, 8</td>
</tr>
<tr>
<td>III</td>
<td>3</td>
</tr>
</tbody>
</table>

$r' = 3$
From Step 2:

<table>
<thead>
<tr>
<th>Subset index</th>
<th>Vertex</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>II</td>
<td>2</td>
<td>(1,0,1)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(0,1,1)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>(0,2,0)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>(0,2,0)</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>(0,2,0)</td>
</tr>
<tr>
<td>III</td>
<td>8</td>
<td>(0,1,1)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(0,3,0)</td>
</tr>
</tbody>
</table>

From Step 3:

<table>
<thead>
<tr>
<th>Subset index</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>2</td>
</tr>
<tr>
<td>III</td>
<td>4, 8</td>
</tr>
<tr>
<td>IV</td>
<td>5, 6, 7</td>
</tr>
<tr>
<td>V</td>
<td>3</td>
</tr>
</tbody>
</table>

$t = 5$. Since $t \neq t'$, then $t' = t$, and repeat Steps 2 and 3.

From Step 2:

<table>
<thead>
<tr>
<th>Subset index</th>
<th>Vertex</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>(0,1,0,0,0)</td>
</tr>
<tr>
<td>II</td>
<td>2</td>
<td>(1,0,0,0,1)</td>
</tr>
<tr>
<td>III</td>
<td>4</td>
<td>(0,0,0,1,1)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>(0,0,0,1,1)</td>
</tr>
<tr>
<td>IV</td>
<td>5</td>
<td>(0,0,1,1,0)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>(0,0,0,2,0)</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>(0,0,1,1,0)</td>
</tr>
<tr>
<td>V</td>
<td>3</td>
<td>(0,1,2,0,0)</td>
</tr>
</tbody>
</table>

From Step 3:

<table>
<thead>
<tr>
<th>Subset index</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>2</td>
</tr>
<tr>
<td>III</td>
<td>4, 8</td>
</tr>
<tr>
<td>IV</td>
<td>5, 7</td>
</tr>
<tr>
<td>V</td>
<td>6</td>
</tr>
<tr>
<td>VI</td>
<td>3</td>
</tr>
</tbody>
</table>

$t = 6$. Since $t \neq t'$, then $t' = t$, and repeat Steps 2 and 3.
From Step 2:

<table>
<thead>
<tr>
<th>Subset index</th>
<th>Vertex</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>(0,1,0,0,0,0)</td>
</tr>
<tr>
<td>II</td>
<td>2</td>
<td>(1,0,0,0,0,1)</td>
</tr>
<tr>
<td>III</td>
<td>4, 8</td>
<td>(0,0,0,1,0,1)</td>
</tr>
<tr>
<td>IV</td>
<td>5, 7</td>
<td>(0,0,1,0,1,0)</td>
</tr>
<tr>
<td>V</td>
<td>6</td>
<td>(0,0,0,2,0,0)</td>
</tr>
<tr>
<td>VI</td>
<td>3</td>
<td>(0,1,1,0,0,0)</td>
</tr>
</tbody>
</table>

From Step 3:

<table>
<thead>
<tr>
<th>Subset index</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>2</td>
</tr>
<tr>
<td>III</td>
<td>4, 8</td>
</tr>
<tr>
<td>IV</td>
<td>5, 7</td>
</tr>
<tr>
<td>V</td>
<td>6</td>
</tr>
<tr>
<td>VI</td>
<td>3</td>
</tr>
</tbody>
</table>

$t = 6$. Since $t = t'$, the module is terminated after three iterations.

Thus the vertex partitioning obtained is

\[ V = V_1 \cup V_3 \cup V_4 \cup V_5 \cup V_6, \]
where $V_1 = \{1\}$, $V_2 = \{2\}$, $V_3 = \{4, 8\}$, $V_4 = \{5, 7\}$, $V_5 = \{6\}$, $V_6 = \{3\}$. It can be noted that in this case, the vertex partitioning generated by module Vertex Parti is identical to the automorphism partitioning of the graph. Thus there are a total of six $^{13}\text{C}$ NMR signals, and the ratio of the intensity pattern is the ratio of the number of elements in the above vertex partitioning, 2:2:1:1:1:1.

As we have mentioned earlier, if a graph is regular and the degree partitioning consists of $V$ alone, then Algorithm I does not refine the partitioning. Obviously here the vertex degree alone, a graph invariant, fails to yield an initial discrimination among the vertices. Since most of the regular graphs of chemical interest, except the $K_2$ graph, are cyclic, another graph invariant one could naturally impose is the number of rings of a specific length containing the vertex in question. See the regular graph I in Fig. 2; for each vertex the number of distinct element rings of each size in which the vertex occurs is calculated by Wipke and Dyott's algorithm (30). This is exemplified below for the graph I in Fig. 2:

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Ring size: 3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

Similarly for graph 2 in Fig. 2 its "ring information" is calculated and listed below:

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Ring size: 3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

Obviously, if we associate each vertex with a ring list $R = (d_3, d_4, \ldots, d_m)$, where $d_i$ is the number of element rings of size $i$ in which the vertex occurs and $m$ is the largest ring size, then in Step 1 of Algorithm I instead of carrying out the degree partitioning, the vertices can be classified into subsets by using their ring lists. It can be stipulated that two vertices belong to the same subset if and only if their ring lists are identical.
Thus for regular graphs with discriminating ring lists the revised Algorithm I should lead to an initial vertex partitioning (possibly an automorphism partitioning) of nuclei set $V$. For graphs 1 and 2 in Fig. 2 the complete automorphism partitioning is readily achieved by such an approach:

**graph 1** \[ V = \{1\} \cup \{2, 6, 9\} \cup \{3, 4, 5, 7, 8, 10\} \]

- number of $^{13}$C NMR signals = 3
- ratio of intensity pattern = 6:3:1 with the number of nuclei in class with unit intensity being 1

**graph 2** \[ V = \{1, 4, 7, 10\} \cup \{2, 3, 8, 9\} \cup \{5, 6\} \]

- number of $^{13}$C NMR signals = 3
- ratio of intensity pattern = 2:2:1 with the number of nuclei in class with unit intensity being 2

On the other hand, however, there are many regular graphs of chemical interests such as the Peterson graph 3 and cubane graph 4 in Fig. 2; their “ring information” alone fails to initially partition the vertices. This is illustrated below with the ring lists of the Peterson and cubane graphs:

**Peterson graph**

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Ring size: 3 4 5 6 7 8 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 6 6 0 12 18</td>
</tr>
<tr>
<td>2</td>
<td>0 0 6 6 0 12 18</td>
</tr>
<tr>
<td>3</td>
<td>0 0 6 6 0 12 18</td>
</tr>
<tr>
<td>4</td>
<td>0 0 6 6 0 12 18</td>
</tr>
<tr>
<td>5</td>
<td>0 0 6 6 0 12 18</td>
</tr>
<tr>
<td>6</td>
<td>0 0 6 6 0 12 18</td>
</tr>
<tr>
<td>7</td>
<td>0 0 6 6 0 12 18</td>
</tr>
<tr>
<td>8</td>
<td>0 0 6 6 0 12 18</td>
</tr>
</tbody>
</table>

**Cubane graph**

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Ring size: 3 4 5 6 7 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 3 0 12 0 6</td>
</tr>
<tr>
<td>2</td>
<td>0 3 0 12 0 6</td>
</tr>
<tr>
<td>3</td>
<td>0 3 0 12 0 6</td>
</tr>
<tr>
<td>4</td>
<td>0 3 0 12 0 6</td>
</tr>
</tbody>
</table>

Here the given “ring list partitioning” consists of $V$ alone; the revised Algorithm I does not refine the partitioning. Thus for a regular graph with nondiscriminating ring lists, we conclude that all the vertices in nuclei set $V$ are equivalent to one another. For such graphs consequently the number of $^{13}$C NMR signals is one, and the intensity of the signal is proportional to the number of nuclei in the vertex set $V$. 
**Algorithm II**

Algorithm II is designed to (possibly trivially) refine the vertex partitioning generated by Algorithm I and eventually give the NMR signal and intensity patterns in the following way.

A spanning tree, with some vertex $i$ under investigation, is called the root. This tree is grown out to the attachments of the starting vertex, and subsequently from these attachments, and so on. Specifically at the $k$th level of the spanning tree, all the attachments of the vertices at the $k$th level which have not been visited at previous levels including the $k$th level are searched. Then these are connected to the attach-

**TABLE 1**

The Spanning Trees and the Valence Vectors for Vertices 5, 6 of Graph 2 in Fig. 1

<table>
<thead>
<tr>
<th>Level</th>
<th>$i$-rooted tree</th>
<th>$V_V$</th>
<th>$j$-rooted tree</th>
<th>$V_{V'}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td><img src="root_tree" alt="Root Tree Diagram" /></td>
<td>(0001)</td>
<td><img src="root_tree" alt="Root Tree Diagram" /></td>
<td>(0001)</td>
</tr>
<tr>
<td>$L_1$</td>
<td><img src="level1_tree" alt="Level 1 Tree Diagram" /></td>
<td>(1300)</td>
<td><img src="level1_tree" alt="Level 1 Tree Diagram" /></td>
<td>(1300)</td>
</tr>
<tr>
<td>$L_2$</td>
<td><img src="level2_tree" alt="Level 2 Tree Diagram" /></td>
<td>(0100)</td>
<td><img src="level2_tree" alt="Level 2 Tree Diagram" /></td>
<td>(0100)</td>
</tr>
<tr>
<td>$L_3$</td>
<td><img src="level3_tree" alt="Level 3 Tree Diagram" /></td>
<td>(0001)</td>
<td><img src="level3_tree" alt="Level 3 Tree Diagram" /></td>
<td>(0001)</td>
</tr>
<tr>
<td>$L_4$</td>
<td><img src="level4_tree" alt="Level 4 Tree Diagram" /></td>
<td>(1200)</td>
<td><img src="level4_tree" alt="Level 4 Tree Diagram" /></td>
<td>(1200)</td>
</tr>
</tbody>
</table>
TABLE 2
The Spanning Trees and the Valence Vectors for Vertices 1 and 8 of Graph 2 in Fig. 1

<table>
<thead>
<tr>
<th>Level</th>
<th>i-rooted tree</th>
<th>V̄V</th>
<th>j-rooted tree</th>
<th>V̄V'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>1</td>
<td>(0100)</td>
<td>8</td>
<td>(0100)</td>
</tr>
<tr>
<td>L1</td>
<td>2 5</td>
<td>(0101)</td>
<td>7 6</td>
<td>(0101)</td>
</tr>
<tr>
<td>L2</td>
<td>7 9</td>
<td>(1100)</td>
<td>5 10</td>
<td>(1201)</td>
</tr>
<tr>
<td>L3</td>
<td>8</td>
<td>(0100)</td>
<td>3 4 10</td>
<td>(1200)</td>
</tr>
<tr>
<td>L4</td>
<td>6</td>
<td>(0001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5</td>
<td>3 4 10</td>
<td>(1200)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ments to their precedent vertices at the kth level to grow out to the next level, i.e., the (k + 1)th level. The edge connecting step is carried out following a priority rule that is specified as follows: among the vertices at the kth level that are adjacent to one common vertex at the (k + 1)th level only the edge between the common vertex and the adjacent vertex with the lowest label at the kth level is connected. The spanning tree is thus called the i-rooted tree; see Fig. 3.

Assume that the vertex set V is partitioned into i subsets by module Vertex_Parti

\[ V = V_1 \cup V_2 \cup \ldots \cup V_i. \]

Consider a subset with more than one vertex in it, say \( V_k \). For two vertices \( i, j \in V_k \), two spanning trees are grown out separately, namely the i-rooted and j-rooted trees.

TABLE 3
The NMR Signal and Intensity Patterns of the Graphs with Isospectral Points in Fig. 4

<table>
<thead>
<tr>
<th>System</th>
<th>n</th>
<th>n (IP)</th>
<th>n (NMR)</th>
<th>Ratio of intensity patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>2</td>
<td>5</td>
<td>1:1:1:1:*</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td>2:2:1:1:1:1:1:*</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>2</td>
<td>7</td>
<td>2:2:2:1:1:1:*</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>1:1:1:1:1:1:*</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>4</td>
<td>10</td>
<td>2:2:2:1:1:1:1:*</td>
</tr>
</tbody>
</table>

Note. \( n \), the number of nuclei in set \( V \); \( n \) (IP), the number of isospectral points; \( n \) (NMR), the number of \(^{13}\)C NMR signals.

* The number of nuclei in class with unit intensity is 1.
*b The number of nuclei in class with unit intensity is 2.
The two growing trees are examined at each frontier level to see if there is any discrepancy between them. The checking starts from the root. At the present frontier level, say the $k$th level of the $i$-rooted tree, a set $A_k$ is constructed, which contains all vertices at the $k$th level. The vertices in the set $A_k$ are themselves the next-nearest neighbors to the corresponding precedent vertices at the $(k - 1)$th level. The set $A_k$ is then degree-partitioned into four disjoint subsets for carbon nuclei,

$$A_k = D_1 \cup D_2 \cup D_2 \cup D_4,$$

such that two vertices belong to the same subset if and only if their degrees are the same. As we mentioned earlier, if nucleus set $V$ contains only carbon atoms, the vertex degrees vary in magnitude from integer 1 to 4. We now construct a $1 \times 4$ row vector called the valence vector, with its components being the number of vertices in each subset $D_i$, $i = 1, 2, 3, 4$, denoted by $V_i$.

$$V_i = (|D_1|, |D_2|, |D_3|, |D_4|).$$

Similarly, at the $k$th level of the $j$-rooted tree, the valence vector $V_j$ is constructed accordingly. The two vectors, $V_i$ and $V_j$, are then compared. If $V_i$ and $V_j$ are not identical for the first time we then conclude that the vertices $i$ and $j$ do not
belong to the same subset, and the spanning tree checking is terminated. If $V_-V$ and $V_-V'$ are identical, the checking is carried on to the $k+1$st levels of both the $i$-rooted and the $j$-rooted trees, and a new set of valence vectors is constructed. These steps are iterated until the final levels have been checked. If valence vectors $V_-V$ and $V_-V'$ are identical from the roots through the final levels, the only conclusion that could be drawn is that the vertices $i$ and $j$ are equivalent to each other. Thus they belong to the same subset under the automorphism partitioning. Let $E_i, i = 1, 2, \ldots, k$ be the subsets of $V$ under the automorphism partition, namely

$$V = E_1 \cup E_2 \cup \cdots \cup E_k = \bigcup_{i=1}^{k} E_i. \quad [6]$$

Let Vertex $-_i$ and Vertex $-_j$ be any two vertices and $j \in V_1$ under investigation. A complete description of the refining Algorithm II discussed above is given in the form of pseudocode:

Module name: VP_Refining
Function: Refining the vertex partitioning from Algorithm I
Get $t, V_i, i = 1, 2, \ldots, t$
$k = 0$
For $1 = 1, t$, do
  if ($|V_i| = 1$) then
    $k = k + 1; E_k = E_k \cup V_1; V_1 = \emptyset$; Go to 10
  End if
  Get next Vertex $-_i \in V_1$
  $k = k + 1; E_k = E_k \cup \{\text{Vertex}-_i\}; V_1 = V_1 \setminus \{\text{Vertex}-_i\}$
  For $m = 1, |V_i|$, do
    Get next Vertex $-_j \in V_i$
    Grow out new levels of spanning trees
    Generate $V_-V$ AND $V_-V'$
    If ($V_-V \neq V_-V'$) Go to 20
    If not the last level, then Go to 111
    $E_k = E_k \cup \{\text{Vertex}-_j\}; V_1 = V_1 \setminus \{\text{Vertex}-_j\}$
  End do
  If ($|V_i| \neq \emptyset$) Go to 10
End do
Output $k, E_i, i = 1, 2, \ldots, k$
End module

Consider the graph 2 in Fig. 1, as an example; the initial partitioning generated by module Vertex_Parti for this graph is

$$V = \bigcup_{i=1}^{3} V_i,$$

where $V_1 = \{9, 10\}, V_2 = \{1, 2, 3, 4, 7, 8\}, V_3 = \{5, 6\}$. 


<table>
<thead>
<tr>
<th>No. of rings</th>
<th>NMR signals</th>
<th>Intensity patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>1:2:2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1:2:2:2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1:2:2:2:2</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1:2:2:2:2:2</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1:2:2:2:2:2:2</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>1:2:2:2:2:2:2:2</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>1:2:2:2:2:2:2:2:2</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>1:2:2:2:2:2:2:2:2:2</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>1:2:2:2:2:2:2:2:2:2:2</td>
</tr>
</tbody>
</table>

* The number of nuclei in class with unit intensity is 2.

![Fig. 5. A linear polyacene graph.](image)

**TABLE 5**

<table>
<thead>
<tr>
<th>No. of rings</th>
<th>No. of $^{13}$C nuclei</th>
<th>NMR signals</th>
<th>Intensity pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>24</td>
<td>3</td>
<td>1:1:2</td>
</tr>
<tr>
<td>19</td>
<td>54</td>
<td>6</td>
<td>1:1:1:2:2:2</td>
</tr>
<tr>
<td>37</td>
<td>96</td>
<td>10</td>
<td>1:1:1:1:2:2:2:2:2</td>
</tr>
</tbody>
</table>

* The number of nuclei in class with unit intensity is 6.

![Fig. 6. A circular polyacene graph.](image)
TABLE 6

<table>
<thead>
<tr>
<th>No. of rings</th>
<th>NMR signals</th>
<th>Intensity pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>1:1:1</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1:1:1:1:1</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>1:1:1:1:1:1</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>1:1:1:1:1:1:1</td>
</tr>
</tbody>
</table>

4 The number of nuclei in class with unit intensity is 6.

For the vertices in subset $V_3$, the complete spanning trees and the corresponding valence vectors are shown in Table 1.

From Table 1 it is seen that vertices 5 and 6 are equivalent to each other, and they should belong to the same subset under the automorphism partitioning. We now consider vertices 1 and 8 in the subset $V_2$. The corresponding spanning trees and the valence vectors are shown in Table 2.

Note that the spanning tree scrutiny is actually terminated at the second level ($L_2$) if the vectors $V' - V$ and $V' - V'$ are not identical for the first time. In this case, we conclude that vertices 1 and 8 are not equivalent to each other, since they do not belong to the same subset under the automorphism partitioning. The complete spanning trees are shown in Table 2 to illustrate this. After repeating the same procedure on the other vertices in $V_2$ and the vertices in $V_1$, the automorphism partitioning of $V$ turns out to be $V = V_1 \cup V_2 \cup V_3 \cup V_4$, where $V_1 = \{9, 10\}, V_2 = \{1, 2, 3, 4\}, V_3 = \{7, 8\}, V_4 = \{5, 6\}$. Therefore, the number of $^{13}$C NMR signals is 4, and the ratio of intensities is 2:1:1:1 with the number of nuclei in class with unit intensity being 2.

The actual computer code was designed such that given a molecule only the number of vertices and the neighborhood information are required as input data. The

Fig. 7. A triangular polyacene graph.
TABLE 7
13C NMR Signal and Intensity Patterns of Polycyclic Compounds with D_{nm} Symmetry in Fig. 8

<table>
<thead>
<tr>
<th>System</th>
<th>NMR signals</th>
<th>Intensity pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>3</td>
<td>1:1:2</td>
</tr>
<tr>
<td>II</td>
<td>5</td>
<td>1:1:1:2:2</td>
</tr>
<tr>
<td>III</td>
<td>5</td>
<td>1:1:2:2:2</td>
</tr>
<tr>
<td>IV</td>
<td>5</td>
<td>1:1:2:2:2</td>
</tr>
<tr>
<td>V</td>
<td>8</td>
<td>1:1:1:2:2:2:2:2:2</td>
</tr>
<tr>
<td>VI</td>
<td>6</td>
<td>1:1:2:2:2:2:2:2:2</td>
</tr>
<tr>
<td>VII</td>
<td>4</td>
<td>1:1:2:2:2:2:2</td>
</tr>
<tr>
<td>VIII</td>
<td>4</td>
<td>1:1:2:2:2:2:2:2:2</td>
</tr>
<tr>
<td>IX</td>
<td>9</td>
<td>1:1:1:2:2:2:2:2:2</td>
</tr>
<tr>
<td>X</td>
<td>8</td>
<td>1:1:2:2:2:2:2:2:2</td>
</tr>
<tr>
<td>XI</td>
<td>7</td>
<td>1:1:2:2:2:2:2:2</td>
</tr>
<tr>
<td>XII</td>
<td>7</td>
<td>1:1:2:2:2:2:2:2</td>
</tr>
<tr>
<td>XIII</td>
<td>7</td>
<td>1:1:2:2:2:2:2:2</td>
</tr>
<tr>
<td>XIV</td>
<td>6</td>
<td>1:2:2:2:2:2:2:2</td>
</tr>
<tr>
<td>XV</td>
<td>5</td>
<td>1:2:2:2:2:2:2:2</td>
</tr>
</tbody>
</table>

* The number of nuclei in class with unit intensity is 6.

modules VP_Refining and Vertex_Partit are called consecutively to generate and refine the vertex partitioning of the nucleus set V. As the final output of the computer code the automorphism partitioning of the nucleus set V provides for the number of NMR signal and intensity patterns.

RESULTS AND DISCUSSION

In this section we discuss many applications of the computer codes developed here to the construction of 13C NMR signal patterns from the neighborhood table.

The first set of molecular graphs we wish to consider is the set of graphs with isospectral points. For any two isospectral points, attachment of a moiety M to these points in turn gives two nonisomorphic graphs, which are often called isospectral graphs. The isospectral graphs are nonidentical graphs but give identical characteristic polynomials and sets of eigenvalues. They have been the topics of many investigations (21-24, 31). We considered many graphs with isospectral points as candidates for the developed code. With no exception the isospectral points were distinguished and the NMR signal and intensity patterns were obtained, as seen from Table 3 (see also Fig. 4).

Table 4 lists the NMR signal and intensity patterns of 13C NMR spectra of linear polyacenes with the rings arranged in a straight line for structures up to 10 rings. Note that naphthalene (number of rings = 2) and anthracene (number of rings = 3) are special cases of this class of polyacenes (see Fig. 5).

Table 5 shows the NMR signal and intensity patterns of some circular polyacenes. The class of circular polyacene structures is generated by starting with a benzene ring and drawing hexagons for every edge of the benzene. For example, the structure with 37 rings, or 96 carbons, is shown in Fig. 6.
Table 6 considers the $^{13}$C NMR of triangular polyacenes of which triphenylene is the simplest example. This class of structures is constructed by starting with a hexagon and choosing alternate edges to construct additional lines of hexagons. The resulting structure has $D_{3h}$ symmetry; see Fig. 7.

In Table 7, we list the $^{13}$C NMR signal and intensity patterns of some polycyclic hydrocarbons with $D_{6h}$ symmetry (Fig. 8).

The developed computer code requires very little CPU time to process various chemical graphs. The average CPU time on a VAX/8650 system is about 0.5 to less than 60 seconds for graphs of size ranging from 3 to 128 vertices. Because each set is represented as a quadruple precision variable that is equivalenced to four integers (32 bits each) in manipulating set operations required by the algorithms, the code can...
handle graphs containing up to 128 vertices. We are, however, optimistic that this code can be extended to other graphs if larger word size is available.

ACKNOWLEDGMENT

This research was supported by the National Institutes of Health under Grant GM37963.

REFERENCES