N.Y., 1972; W. H. Guertin and J. P. Bailey, Jr., "Introduction to Modern Factor Analysis", Edwards, Ann Arbor, Mich., 1970; D. N. Lawley and A. E. Maxwell, "Factor Analysis as a Statistical Method", American Elsevier, New York, 1971; W. W. Cooley and P. R. Lohnes, "Multivariate Data Analysis", Wiley, New York, 1971, pp 96–167; D. F. Morrison, "Multivariate Statistical Methods", 2nd ed, McGraw-Hill, New York, 1976, pp 266–343; A. S. C. Ehrenberg, "Data Reduction", Wiley, New York, 1975, pp 263–282; K. Enslein, A. Ralston, and H. S. Wilf, Ed., "Statistical Methods for Digital Computers", Vol. 3, Wiley, New York, 1977, pp 9–10, 123–202; E. Weber, "Einführung in die Faktorenanalyse", VEB Gustav Fischer Verlag, Jena, 1974.

(12) N. H. Nie, C. H. Hull, J. G. Jenkins, K. Steinbrenner, and D. H. Bent, "Statistical Package for the Social Sciences", 2nd ed, McGraw-Hill, New York, 1975, pp 468–514.

(13) W. J. Dixon, "BMD Biomedical Computer Programs", University of California Press, 1973; W. J. Dixon, "BMPD Biomedical Computer Programs", University of California Press, 1975.

(14) D. G. Armor and A. S. Couch, "Data-Text Primer", Macmillan, New York, 1972, pp 82–91.

(15) R. Buhler, "P-STAT", Princeton University, 1975.

(16) "System/360 Scientific Subroutine Package Programmer's Manual, Version III", International Business Machines Corp., Manual GH20-0205-4, 1968.

(17) "IMSL Library 1" (IBM Series), 5th ed, 1975; and "IMSL Library 2" (Honeywell Series), 4th ed, 1974, International Mathematical and Statistical Libraries, Houston, Texas.

(18) B. H. Hall, "Time Series Processor, Version 2," Harvard Institute of Economic Research, Harvard University, Cambridge, Mass., 1976.

(19) L. L. Thurstone, "Multiple-Factor Analysis", University of Chicago Press, Chicago, Ill., 1947, pp 117–124.

(20) The eight synthetic problems or "artificial experiments" listed in ref 10, pp 528–9.

(21) Four SPSS extraction procedures (PAl, PA2, Rao, and alpha) with six rotations (varimax, equimax, quartimax, and oblique with $\delta = -1, 0, +1$) were carried out. Image extraction failed (singular correlation matrix) even with 0.1–1% data errors (absolute magnitude changed by 1 in third significant figures, up for odd $i + j$, down for even $i + j$). None of the combinations gave realistic sets of factors. Quartimax also made b_2 positive and b_7 negative. Highly oblique ($\delta = +1$) structure factors were only slightly negative for b_2 but varied more than threefold from b_4 to b_7. Most other sets had numerous incorrect signs.

(22) One-mode problems, such as the chemical one of best fitting the Hammett equation with a single type of substituent constant (S. Wold and M. Sjöström, *Chem. Scr.*, **2**, 49 (1972); S. Wold, *ibid.*, **5**, 97 (1974); M. Sjöström and S. Wold, *ibid.*, **6**, 114 (1974); **9**, 200 (1976)), do not lead to such errors and confusion, because the number of critical conditions is zero for them; hence there is no possibility of choosing the critical conditions incorrectly.

Graph-Based Fragment Searches in Polycyclic Structures

MILAN RANDIĆ*[1] and CHARLES L. WILKINS*

Department of Chemistry, University of Nebraska—Lincoln, Lincoln, Nebraska 68588

Received May 31, 1978

A procedure for substructure search in polycyclic systems based upon the concept of atom codes and their comparison is outlined. The atom codes used represent the number of paths of different length originated at each atom. This scheme represents an extension to polycyclic structures of a recently described algorithm for fragment searches in acyclic structures. The basic idea of using the number of neighbors and number of paths for characterization of an atomic environment is not unfamiliar. However, it appears that the accompanying formalism has not been developed previously. The path concept has found much application in various problems relating graphs and structures, but previously the primary emphasis has been the search for shortest paths. Here we show it is useful to list all paths (to any desired depth) for all atoms in a structure and then to use such a list for substructure searches. Essentially, comparison of the lists of paths for two graphs permits establishing compatibilities among the vertices and results in assignment of multiple labels to the graph under investigation, where the labels are those of the fragment. Subsequent atom-by-atom matching is straightforward and efficient, since the use of multiple labeling prevents a large number of otherwise unproductive searches from even being considered. By successive registration of a fragment, vertices whose labels have been exhausted are deleted and the process repeated on a smaller graph with fewer multiple labels—which makes possible rapid convergence in the search. This search procedure is illustrated with an example.

INTRODUCTION

The problem of fragment searching or, as it is known among mathematicians, the problem of subgraph isomorphism, continues to attract considerable attention in diverse scientific disciplines. It appears under different guises in its special forms (e.g., the search for the shortest path or a search for a Hamiltonian circuit). In chemistry, the problem is of interest not only to those involved in chemical documentation and the use of computers in searching for optimal synthetic routes but also has possible implications for pattern recognition and other manipulations utilizing large files of data, as well as for drug design and/or structure–activity correlations. The continuing development of experimental techniques and the rapid accumulation of more reliable and more accurate experimental data mandates improved algorithms for computer manipulation of the results.

Comparison of structural representations and their correlation with associated data requires efficient fragment search algorithms. Here we are primarily interested in substructure searches, where the term implies a collection of connected atoms, rather than searching for a substructure identified by a conventional (or trivial) name such as is used in substructure searches of an index, for example. The input information in our case is the connection table for atoms forming the fragment, or pictorially, the graph corresponding to the fragment. The problem of subgraph isomorphism has been considered in the literature, though not so thoroughly as the related problem of graph isomorphism.[2] Conceptually, the simplest and probably the oldest scheme is atom-by-atom matching which is equally suitable and, in application, practically identical whether one searches for graph isomorphism or subgraph isomorphism.[3] In an atom-by-atom search, one proceeds from the selected atom to its neighbors sequentially selecting single bonds and continues the process as long as matches are found. As soon as a nonmatch occurs, one back-tracks to the previous branching point and selects an alternative bond. The search is continued until the desired fragment is found or until all alternative routes have been exhausted. Besides requiring back-tracking, which is wasteful, the scheme also requires extensive bookkeeping, and even for

0095-2338/79/1619-0023$01.00/0

molecules and fragments of modest size the search soon becomes impractical. In addition, in polycyclic structures, one usually must restrict the number of atoms and bonds because of possibility of registering "false" chains within ring systems.[4] One can considerably improve the search speed by prior classification of atoms or atoms and bonds including information concerning the type of atoms, the number of neighbors, the kind of bonds, and so on. Such an approach characterizes the methods of both Figueras[5] and Saucier[6] and is also the basis of the procedure favored by Sussenguth[7] which is the first alternative scheme proposed in the effort to find a practical answer to the deficiencies of atom-by-atom matching. Finally, classification of atoms (vertices) is also the basis of a more general problem than locating subgraph isomorphism, that of searching for the maximal common subgraph within two given graphs.[8] In a recently published outline of the present search for subgraphs in acyclic structures,[9] one of us also classified vertices according to the number of neighbors a certain distance away and found that the scheme leads to all the fragments through a rather efficient search. The criterion of efficiency here is the number of unproductive searches which are subsequently discarded. When these are absent or few, the method is considered efficient.

In the present paper, we extend this scheme to polycyclic structures which have been excluded previously because of their additional complexity and a number of other unresolved problems. It is obvious that if a scheme is going to be of interest in application, it ought to apply equally well to acyclic, cyclic, and polycyclic structures. Although in acyclic structures enumeration of the neighbors a certain distance away is *unique* (since between any pair of vertices there is a unique path), in polycyclic structures there are multiple paths and one has to resolve the multiplicity of options. The extension of a scheme from acyclic structures to cyclic and polycyclic is, therefore, by no means trivial or straightforward. We report here how we resolved this issue and devised a procedure to arrive at unique atom codes for cyclic structures. These codes were subsequently used in a manner analogous to that employed when searching for fragments contained within an acyclic system.

FORMULATION OF THE PROBLEM AND DIFFICULTIES ENCOUNTERED

The general graph isomorphism problem is difficult in the sense that comparison of all possible alternative labelings of the graph involves $n!$ cases for graphs having n vertices. As early as 1736, Euler[10] in his discussion of the problem of Königsberg's bridges discards listing all possibilities as an acceptable solution. One has to do better, and solutions can be evaluated and judged in the spirit of the complexity theory by examining the character of the algorithm, which can be exponential or polynomial. In the latter case, one can compare solutions simply by the comparing the power of the polynomials involved. As is known,[11] graph isomorphism and subgraph isomorphism are in the same class complexity problems, though the latter is generally considered more difficult. The problem of identifying graph isomorphism is solved once we succeed in relabeling vertices of the two graphs being compared so that they correspond to the same form of the adjacency matrix. The difficulty of the problem is in finding rules to govern such labeling which would apply equally to all situations and to find an implementation of the rules which does not require an excessive search time. Some progress in this direction has been made by consideration of canonical numbering of atoms based on a particular interpretation of the adjacency matrix.[12] However, having a scheme for a unique labeling of vertices does not help in the subgraph isomorphism problem. The problem expressed via vertex labels becomes that of identifying

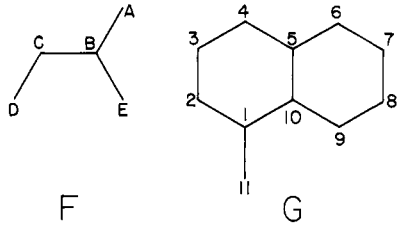

Figure 1. A methylbutane fragment (F) and naphthyl-type skeleton (G).

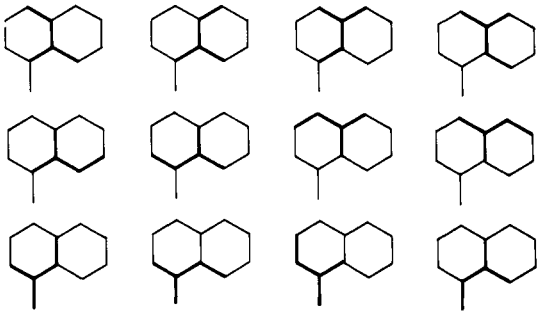

Figure 2. Methylbutane fragments in a naphthyl-type skeleton.

all acceptable labelings in one graph which preserve the adjacency relationship found in another (fragment) graph. For example, consider the fragment F and the graph G of Figure 1. There are, in all, 12 possible ways the fragment F may appear in structure G (Figure 2). If we are interested in obtaining all the possible fragment locations in order to determine how many ways a fragment may appear via vertex labeling, we would have to produce for each case (acceptable fragment found) a special numbering. The advantage of the vertex code concept[9] lies in the possibility of using multiple labeling for G based on the compatibility of the atom codes of G and those of F. However, to be able to proceed in that direction, we need atom codes for polycyclic structures. No simple coding scheme has yet been reported in the literature for polycyclic atomic environments. The classification schemes for atoms (previously mentioned) provide various ways of arriving at discriminatory decisions and differentiating various atoms and atom environments, but do not appear entirely satisfactory. Furthermore, any list of invariants used for comparison which works on cases already tested may well fail when a novel situation arises and a structure not anticipated appears. In the present approach, based on the concept of atom codes, we advocate a somewhat different philosophy: atom codes serve for classification of vertices, but by no means do we require that each class contain only one item. It is desirable if the list of atom codes is unique for each structure, but single codes do not characterize the atom connectivity uniquely. Moreover, nonequivalent atoms within the same structure may have the same code. Atom codes primarily serve to maximally partition the atom set, but subsequent examination of connectivity in what is similar to an atom-by-atom search confirms isomorphism, finds a fragment, or conclusively establishes that no such fragments exist. In contrast, however, to the atom-by-atom matching procedure one explores the neighborhood of atoms only in those directions which, prior to the search, have been found to qualify.

ATOM CODES

Although the concept of enumerating atom neighbors located a certain distance apart and the idea of path enumeration are already familiar and have been used in different situations, it seems that such codes have not been fully developed and formally treated as mathematical entities. Our emphasis here is on this mathematical formalism. Such an attitude con-

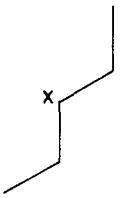

Figure 3. An atom X possessing the same code (2,2) as atom C in Figure 1, but a different environment.

siderably broadens the potential applicability and also makes the codes available for both algebraic and logistic manipulations. Here as a *code* we understand a *sequence of integers* derived by *enumeration* of a selected *graph invariant.* Such sequences can be *added* to produce new codes; they can be *manipulated* by erasure of some entries and the shift of others a single place to the left or right; they can be *altered* in such a way that a new code corresponds to a different (acceptable) structure or atomic environmentf or the codes can be *added to other sequences* or *even multiplied*, generating quantities and new graph invariants. Some applications of such codes in other problems, such as examination of structure–property correlations,[13] have been considered elsewhere.

For acyclic structures, atom codes have been defined by enumeration of neighbors (for each atom separately) one, two, three, etc., bonds away. For the fragment, F, we have the following codes:

A	1,2,1
B	3,1
C	2,2
D	1,1,2
E	1,2,1

Equivalent atoms A and E necessarily have the same codes. Codes of nonequivalent atoms will generally differ, but occasionally the same code may appear for nonequivalent atomic environments. This is not surprising, since we record only the *number* of atom neighbors, and not their distribution in various branches. Thus, the code 2,2 besides the environment of atom C also characterizes environment X in Figure 3. Now, consider the cyclic graph G (Figure 1). Enumeration of neighbors now becomes ambiguous since there are multiple paths available for traversing from one atom to another. Some additional qualification is essential in order to make the derivation of codes unique. One could, for example, enumerate neighbors, disregarding which path has been used when there are two or more paths of the same length, and record neighbors only when the shortest paths are used. For atom 1 of the graph G, we would then have:

distance: 1 bond	neighbors: 2, 10, 11	total: 3
distance: 2 bonds	neighbors: 3, 5, 9	total: 3
distance: 3 bonds	neighbors: 4, 6, 8	total: 3
distance: 4 bonds	neighbors: 7	total: 1

The final code for atom 1 in this case is 3,3,3,1. Such codes have been used in an attempt to classify carbon-13 chemical shifts in benzenoid systems.[14] Observe that, in the process of enumerating neighbors three bonds apart, we arrived at atom 4 via two routes (1-2-3-4 or 1-10-5-4), but for atoms 6 and 8 we have only one (shortest) route. This kind of information is lost once the code is constructed. Polycyclic benzoid systems represent regular structures (all rings are hexagonal and the molecules are planar), and the simple codes appear to preserve sufficient useful information to enable one to search for regularities in available experimental data. In more general polycyclic structures, it would be desirable to preserve, initially, as much as possible of the original structural information. For this purpose it appears that enumeration of *paths* rather than *neighbors* offers better possibilities. *Path* is defined in graph theory as a sequence of edges and vertices (or simply as sequence of vertices *or* sequence of edges) which are joined (i.e., adjacent to one another) and in which no vertex is used twice. For atom 1 of graph G, we have the following paths of different length:

length	path	no. of paths
1	1-2; 1-10; 1-11	3
2	1-2-3; 1-10-5; 1-10-9	3
3	1-2-3-4; 1-10-5-4; 1-10-5-6; 1-10-9-8	4
4	1-2-3-4-5; 1-10-5-4-3; 1-10-5-6-7; 1-10-9-8-7	4
5	1-2-3-4-5-6; 1-2-3-4-5-10; 1-10-5-4-3-2; 1-10-5-6-7-8; 1-10-9-8-7-6	5
.	. . .	.

and so on. The corresponding code for atom 1 is then 3,3,4,4,5 . . . . In Table I, truncated atom codes for vertices of graph G are listed. For smaller molecules with at most two rings (bicyclic) such as in the example discussed, the enumeration of paths is not difficult and can be derived from inspecting the molecular diagram. Introduction of another ring (not so much enlargement of existing rings with additional atoms) makes visual enumeration almost impossible, unless truncated codes of relatively short length, which exclude much of the multiple cyclic environment for each atom, are considered. However, it is possible to find all paths, even in these more complex situations, *provided a systematic registry of existing paths is maintained.* For that purpose, it is useful and convenient to introduce *path graphs* which represent the searching tree for paths. We will illustrate path graphs in the following section in which an actual fragment search will be described for somewhat larger molecular structures and fragments. Elsewhere we have described an algorithm for finding all paths (ALLPATHS)[15] which has been programmed in PL/1, Basic, and Fortran. In that paper, limitations (restrictions) and other qualities of the path search algorithm are given. The number of paths increases with the size (i.e., the number of vertices) of a graph and even more so with an increase in the number of edges per vertex (and increase in the number of cycles) so that listing all paths for a rather large graph may become impractical. However, for the problem

Table I. Atom Codes for the Carbon Skeletons of Methylnaphthalene, Dimethylcyclohexane, and Pimarane

Naphthyl			Pimarane	
Atom	Code		Atom	Code
1	3,3,4		1	4,3,5,5,9,9
2	2,3,3		2	2,4,3,7,8,11
3	2,2,4		3	2,2,6,6,10,7
4	2,3,4		4	2,4,5,10,7,12
5	3,4,5		5	4,5,8,6,13,8
6	2,3,4		6	3,6,6,12,9,11
7	2,2,3		7	2,3,8,7,11,8
8	2,2,3		8	2,4,4,7,8,12
9	2,3,5		9	4,3,3,5,10,9
10	3,5,4		10	2,5,5,5,7,15
11	1,2,3		11	3,4,8,8,15,7
			12	2,3,5,13,10,12
Dimethylcyclohexane		Code Sum	13	2,3,8,7,12,9
A	1,3,2,2,2,2	12	14	3,7,5,7,8,16
B	4,2,2,2,2	12	15	1,3,3,5,5,9
C	2,4,2,2,2,2	14	16	1,3,3,5,5,9
D	2,2,4,2,4	14	17	1,3,5,8,6,13
E	2,2,2,6,2	14	18	1,3,3,3,5,10
			19	2,3,2,3,4,9
			20	1,1,3,2,3,4

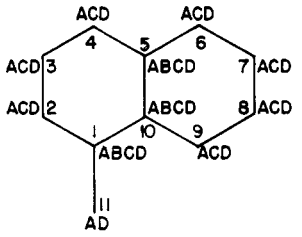

Figure 4. Multiple labeling of graph G.

in which we are interested here—fragment searches—we need only to keep a record of paths of the length determined by the fragment, which is a fraction of the maximal path length in most graphs investigated. In fact, one need not be inflexible about the truncation of codes, and, in some instances, optimal searches can be made with relatively short (truncated) codes. This is possible because the primary purpose of atom codes is to maximally discriminate between various atom environments and usually a four- to five-term code will suffice.

BRIEF OUTLINE OF THE SEARCH

Once we have atom codes for the fragment and a polycyclic graph, the search proceeds quite analogously to the search for a fragment in acyclic structures. Since this has been fully described previously, we will only briefly indicate the necessary steps, before applying the technique to a larger molecule and fragment in the next section. For this purpose, continue with the fragment F and the graph G. Since vertices A and E in the fragment are equivalent, we will no longer use label E. Such simplifications are not essential and if one overlooks apparent symmetry, no error will result, only more labor. The symmetry of graphs should be used maximally when known or apparent. It may pay in some complicated situations to explore symmetries of the underlying graphs, although the task itself is not always simple.[16] The first search step consists of comparing the atom codes of the fragment with the atom codes of the graph (which, in this example, can be truncated to three entries only) in order to see which graph vertices are *compatible* with which fragment vertices. For a graph vertex x to be comparable with fragment vertex X, the former must have all entries equal or larger than the corresponding entries of the latter. If that were not the case, the former could not have an equivalent environment and would be *missing* some of the neighbors and paths characterizing the latter. When a vertex in G is compatible with a vertex in F (fragment), it takes its label. In this way after all vertices are examined, we arrive at a multiply labeled graph G (Figure 4). Vertex 1 has all labels, i.e., can be, potentially, any of the vertices of the fragment. Vertex 2 similarly has several labels, but it cannot represent vertex B of the fragment. Such a labeling process may result in some vertices having only a single label or some vertices remaining unlabeled. In the example considered, vertex 11 has the smallest number of labels, which signals that this particular vertex is most suited for continuation of the search. It should be noted that it is conceivable that a situation may arise where either all vertices have all labels or one of the labels of the fragment does not appear. Such extreme cases would arise if the selected fragment is too small (relative to the structure) and is present all over the structure. Alternatively, if the fragment is too large and is absent from the structure, one will also encounter this situation. The former case, if of continued interest, can be resolved by selecting any of the vertices and gradually reducing the size of the investigated graph with an accompanying increase in the efficiency of the search.

For the example above of the fragment F, *we select vertex 11 and label*, say A, and start listing possible fragments. Adjacency in the fragment *requires* that label A has neighbor

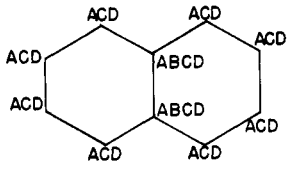

Figure 5. Labeling of the reduced graph.

B. *This condition is also satisfied* for the multilabeled graph G. So we continue; B in the fragment has *two* neighbors: A and C. Since both vertices 2 and 10, which are adjacent to 1, have labels A,C, we have here two possibilities:

(11-A), (1-B), (2-A), (10-C) and
(11-A), (1-B), (10-A), (2-C)

Each of the possibilities has to be further examined, which calls for searching for adjacent vertices with label D to be joined to the already assigned vertex C. As is easily seen, both of the two alternatives meet this last condition. In fact, the first alternative generates two possibilities, since vertices 5 and 9 are D. Hence, we have found two fragments:

(11-A), (1-B), (2-A), (10-C), (5-D); and
(11-A), (1-B), (2-A), (10-C), (9-D)

and another fragment from the other possibility:

(11-A), (1-B), (10-A), (2-C), (3-D)

This completely exhausts label A of vertex 11. At this point, if the object of the search is to determine whether the fragment existed within G, we would halt the search. If the object is to determine how many ways the fragment can be found, we continue. Vertex 11 can also represent vertex D of the fragment so, to continue, such possibilities must now be fully examined. If 11 represents D, then the adjacent vertex (1) must have the label C, which is indeed the case. Next to C is vertex B in the fragment. So, in graph G, we search for an adjacent B and find it as vertex 10. Finally, B has *two* adjacent A vertices and the same condition is also met in the graph (vertices 5 and 9), so that we have found yet another fragment:

(11-D), (1-C), (10-B), (5-A), (9-A)

This completely exhausts vertex 11; i.e., we have found all fragments that include this particular atom. For continuation of the search, we can erase vertex 11 and consider the reduced graph (Figure 5). This reduced graph (naphthalene) has some symmetry and that fact could be exploited in the continuation of the search. This observation also suggests that if there are several alternative vertices to choose for examination, one should prefer those which, when eliminated, will produce a graph with some symmetry. Truncated codes for naphthalene are:

1. 2,3,4
2. 2,2,3
10. 3,4,4

Codes for other vertices can be derived from those given by symmetry arguments. Not much has changed in the assignment of multiple labels, except that vertex 1 is no longer compatible with fragment vertex B. The appearance of multiple labels only indicates the possibility that such and such a vertex of graph G may also be a vertex of fragment F, but to verify this one has to *examine connectivity* as outlined above. For example, in the naphthalene multiple-labeled skeleton, we can immediately see that atom 2 (and, because of the symmetry, also atoms 3, 7, 8) cannot represent vertices A and C of the fragment F although its code is compatible with codes A,C, because the connectivity at A and C requires an adjacent B, and atom 2 has no adjacent vertex B. So we can continue with the search using a graph with multiple labels as shown

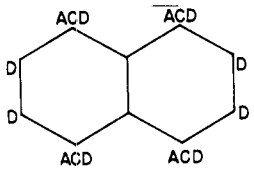

Figure 6. Further reduction of labeling of graph G.

in Figure 6. The continuation may proceed by now selecting vertex 2 and examining its involvement. Even if we have not spotted the inadmissibility of labels A and C for vertex 2, that would now be immediately recognized. In a general situation, one may nevertheless prefer to intercept the search with this additional screening which may help in selecting the next atom to be examined and influence search speed.

AN ILLUSTRATION

The example discussed is too simple to impress upon the complexity of searches that are more likely to appear in practical applications and, on the other hand, does not introduce various situations which arise in larger more difficult problems. We will now fully document a search for a fragment in a larger system and provide the reader with sufficient information that she or he may implement the scheme. Admittedly, we still limit our examination to systems in which all skeleton atoms are the same (hydrocarbons) and all the bonds are same, but such restrictions can be lifted simply by introduction of additional qualifiers for atoms and bonds. In fact, the present problem is more difficult, since additional labels (such as indicators for heteroatoms) will only further restrict the compatibility of codes, which will have to satisfy *besides* connectivity constraints (as dictated by the atom codes) other chemical constraints (bond type, atom kind). We plan to continue to develop this particular path searching and atom code-based scheme for more general structures (which also introduce stereochemical considerations). Although development of such a scheme will require time and experience, it seems to us that some underlying features of our approach may already be of interest to others at this stage and may be applicable to other problems. We have selected the carbon skeleton of pimarane (a diterpene) as an appropriate example. The fragment to be sought is the dimethylcyclohexane skeleton. The fragment (dimethylcyclohexane) atom codes can be derived (Table I) by inspecting the molecular skeleton. We have indicated the sum of paths for each atom. For monocyclic systems having n atoms, the sum of all paths for an atom is equal or less than $2(n-1)$. The maximum value belongs to ring atoms without substituents, while exocyclic pendant vertices have as a sum of paths a value decreased from the maximal by n', where n' is the number of exocyclic atoms.[17] The same reduced value also occurs for ring atoms bearing substituents.

The graph investigated (the skeleton of pimarane) is too large to afford derivation of atom codes simply by inspection, and one has to proceed systematically. Consider atom 1 (Figure 7) and list its first neighbors: 2, 14, 15, 16. Next, take the first of these and search for its first neighbors and continue the process until all paths have been exhausted for the first in the list of several neighbors. Then, return to the previous branching site and explore the possibilities due to other nearest neighbors at the branching point. In this way, gradually all paths will be recorded. For atom 1, we would have at the beginning the following string of paths:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
	14				14	11			18				
	15				17				19				
	16												

When multiple possibilities exist, we first select the neighbor

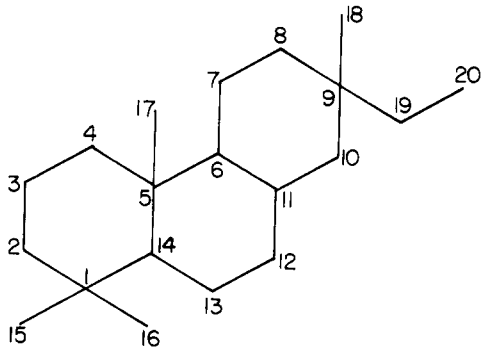

Figure 7. Pimarane skeleton.

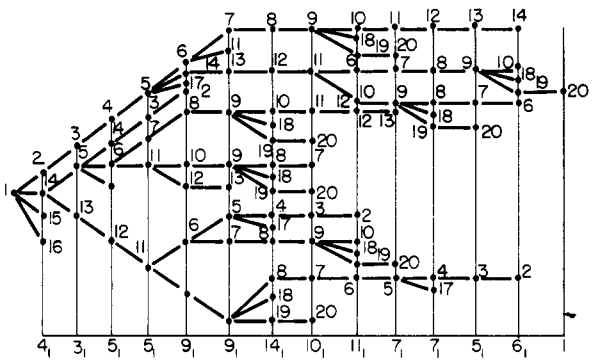

Figure 8. Path graph for vertex 1 of the pimarane carbon skeleton.

with the smallest label. After atom 14, as the neighbors we have 1 and 5; but these atoms have already been visited (i.e., appear already in the path list); hence the path is terminated. One goes back to the preceding branching point (atom 9) and continues to explore paths that branch here. Atom 18 is terminal, but from 19, one can proceed to 20, which terminates the path. Next, we return to the previous branching site (atom 6) and continue exploring possibilities there and so on. A summary of such an inspection of all possibilities originating from atom 1 is depicted in the form of a search tree in Figure 8. The derived acyclic graph contains all paths of atom 1; hence, we refer to such graphs as *path graphs*. For example, if one wishes all paths of length five, one reads these from Figure 8:

1	2	3	4	5	6	
1	2	3	4	5	14	
1	2	3	4	5	17	
1	14	5	4	3	2	and so on

Since we are interested only in the *number* of paths of each length, we have to enumerate the number of vertices in each vertical line. Then, since the fragment has as its longest path one of length six, we can truncate construction of path graphs after the sixth entry. Figure 9 contains truncated path graphs for selected vertices of the graph investigated, while Table I lists the truncated codes. Once we have derived path graphs for the original graph, the same path graphs will be of help in deriving atom codes for reduced graphs later to be considered when one successively erases vertices that have been exhausted in the search. For new codes, one simply ignores the selected vertices in all path graphs and reenumerates the paths. For example, the six-length code for atom 1 is (Figure 8) 4,3,5,5,9,9. If, in the process of the search, atoms 15 and 16 are eliminated, we ignore their presence in path graphs and obtain for the code 2,3,5,5,9,9. However, if for instance, atom 4 is erased, we have to eliminate from the path graph not only atom 4 (which appears twice at length three) but also all vertices that follow after it. The new atom 1 code then becomes 4,3,3,3,5,6.

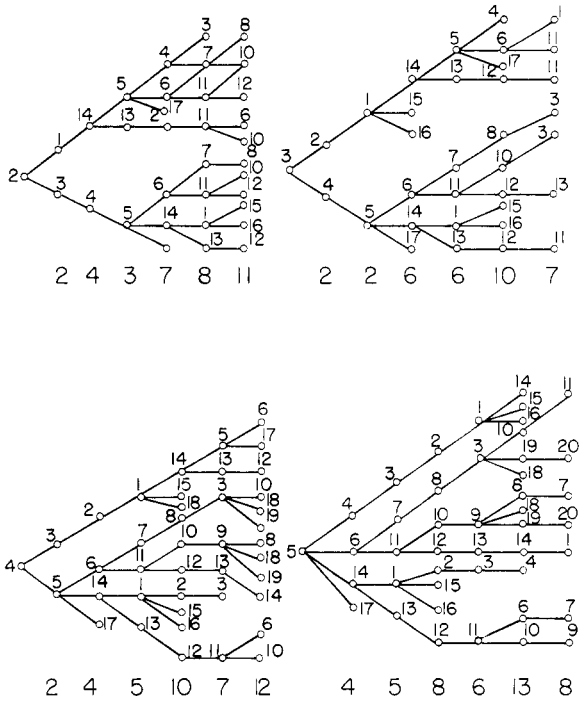

Figure 9. Truncated path graphs for selected vertices of (pimarane).

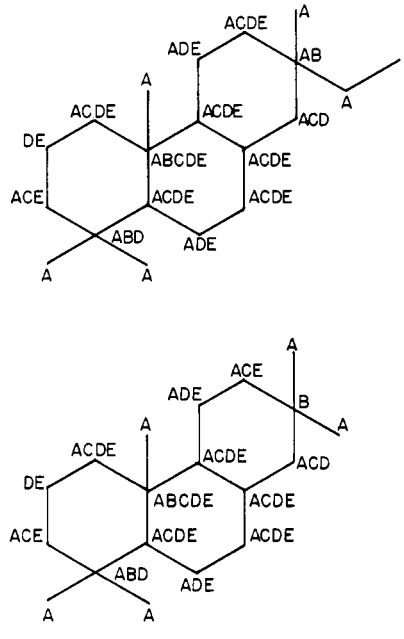

Figure 10. Pimarane graph with multiple labels: (a) original graph with vertex 20 being unlabeled; (b) reduced graph after elimination of vertex 20. Notice some change of labeling in the neighborhood of the erased vertex.

Figure 10 is the graph of pimarane with multiple labels. Since vertex 20 is not compatible with any of the fragment vertices, it can immediately be erased. This erasure only affects labeling of vertex 8 which no longer is compatible with label B. Observe how the procedure of comparison of graph codes differentiates among different vertices, restricting assignment of fragment atoms. Inspection of Figure 10 and verification for adjacency present in the fragment would further eliminate some of the labels. For example, vertex E in the fragment is adjacent to two vertices D. However, in Figure 10 at site 3, the condition is not fulfilled. Therefore, atom 3 cannot be vertex E of the fragment. Similarly, label A must always be adjacent to label B and this is not the case for vertices 7, 11, 12, and 13. Additional such eliminations

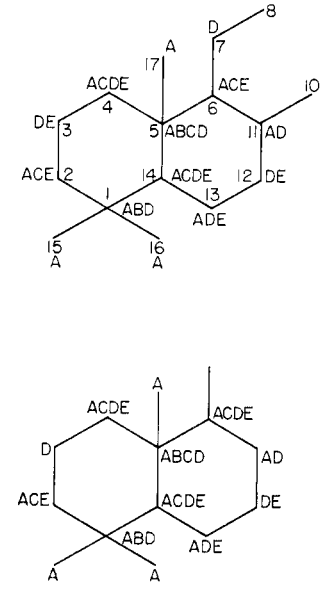

Figure 11. Mutilated graph obtained by erasure of vertices 9, 18, and 19 after the first fragment has been found and new multiple labels: (a) vertices 8 and 10 remain unlabeled; (b) graph and labeling after removal of vertices 8 and 10. Notice that now vertex 7 remains unlabeled.

of labels can be found, and they indicate that the already multiply labeled graph has fewer possibilities than it may appear from only counting the labels shown. Elimination of redundant labels can be accomplished through the process of exhaustion (which, in fact, is part of such a preliminary analysis), when a particular vertex is considered.

Once all vertices are labeled, an actual search starts by selecting a vertex to initiate the search. We will select 19-A as the beginning, which demands the assignment 18-B. Since B has two C vertices adjacent, as well as an A, this requires that the choice of A as a label for 8 and/or 10 must not be made since such a selection does not provide for one of the C assignments. Thus, we have 18-A, 8-C, and 10-C. Each C must have an adjacent label D. Indeed, this condition is fulfulled. Finally, both D vertices require an adjacent vertex E. This is the case with vertex 6, but not for vertex 12. Thus, we have found a fragment:

(19-A), (18-A), (9-B), (8-C), (10-C), (7-D), (11-D), (6-E)

or more concisely:

6,7,8,9,10,11,18,19

(where we have indicated only the vertices involved). Now, we can eliminate from further consideration vertices 9, 18, and 19, since they each had only a *single* label. In each case, the label has been consumed by listing the underlying fragment. In the next step (Figure 11), we consider the graph in which vertices 9, 18, and 19 have been deleted and repeat the process. The first step is to find new atom codes and compare them with the codes of the fragment to verify compatibiliy. The new codes can be obtained quickly and simply from the *path graphs* already found (shown in Figure 9) by deleting atoms 9, 18, and 19. Because 9 is not a terminal vertex, its erasure also implies erasure of all vertices that follow behind it as illustrated in Figure 12 for vertices 7 and 8.

As seen in Figure 11, the assignment of the multiple labels leaves vertices 8 and 10 without labels, indicating that these vertices cannot be part of fragment F. Their deletion similarly produces a graph with a single unlabeled vertex (see site 7 in the lower part of Figure 11) so that without any node-to-node matching, this vertex can also be eliminated from further

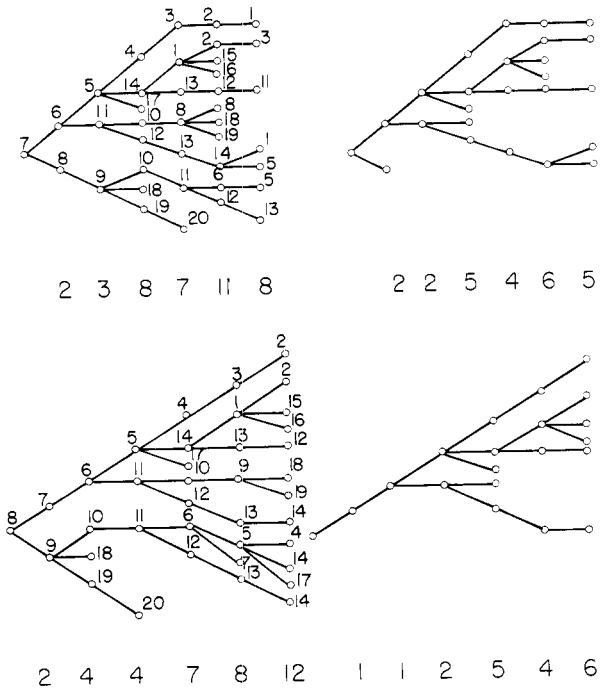

Figure 12. Path graphs and the corresponding truncated codes for selected vertices of the original graph and graph derived by erasure of vertices 8, 18, 19.

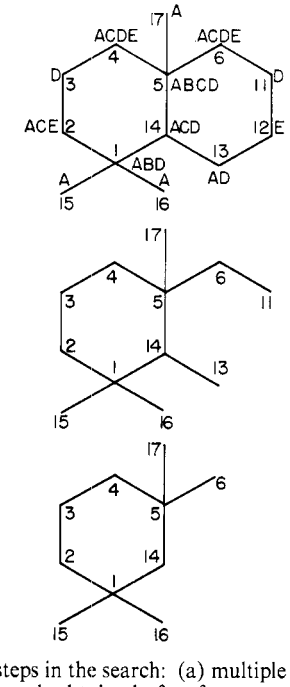

Figure 13. Final steps in the search: (a) multiple labels after removal of vertex 7; (b) graph obtained after fragment involving vertex 12 has been found and the vertex deleted; (c) graph left after removal of unlabeled vertices of the preceding step.

analysis. So we arrive at the reduced graph of Figure 13 with all vertices having a single or multiple labels, and indicating compatibility with fragment codes. The compatibility with fragment codes is demonstrated in Table II. From here, one continues the search for fragments as already described. We can start with vertex 12 assigned as E and proceed to check connectivity for the fragment, first looking for two adjacent D labels and so on. We obtain the fragment:

(12-E), (13-D), (11-D), (14-C), (6-C), (5-B), (4-A), (17-A), or 4-5-6-11-12-13-14-17

No other possibility emerges. Hence, vertex 12 is consumed

Table II. Atom Codes for the Reduced Skeletons of Figure 13 and Their Compatibility with the Fragment Codes

ATOM	CODE						COMPATIBILITY				
1	4	3	5	4	6	4	A	B	-	D	-
2	2	4	3	7	6	6	A	-	C	-	E
3	2	2	6	5	8	4	-	-	-	D	-
4	2	4	4	8	5	5	A	-	C	D	E
5	4	4	6	4	6	2	A	B	C	D	-
6	2	4	4	6	5	9	A	-	C	D	E
11	2	2	4	5	11	9	-	-	-	D	-
12	2	2	3	4	6	6	-	-	-	-	E
13	2	3	7	4	6	4	A	-	-	D	-
14	3	7	4	4	4	4	A	-	C	D	-
15	1	3	3	5	4	6	A	-	-	-	-
16	1	3	3	5	4	6	A	-	-	-	-
17	1	3	4	6	4	6	A	-	-	-	-

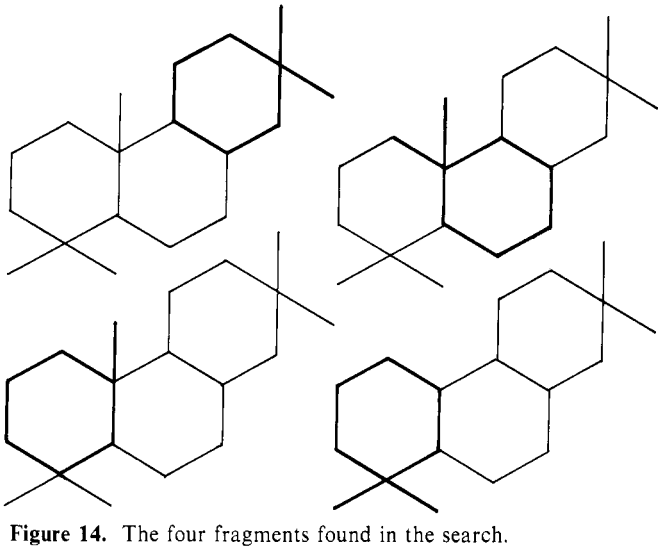

Figure 14. The four fragments found in the search.

and can be eliminated. However, vertex 17 should not be erased just because it has a *single* label. We have not yet examined vertex 17, and it is possible that it participates in several fragments, although *always* as vertex A. In fact, as will be seen shortly, this is the case. The new reduced graph (Figure 13b) has several vertices without labels which, after elimination, lead to the graph of Figure 13c, which immediately gives to additional fragments:

(17-A), (6-A), (5-B), (4-C), (14-C), (3-D), (1-D), (2-B), or 1,2,3,4,5,6,14,17

(15-A), (16-A), (1-B), (2-C), (14-C), (3-D), (5-D), (4-E), or 1,2,3,4,5,14,15,16

This completes the search; we have found four possible fragment matches (Figure 14).

DISCUSSION

From the example and description of the search, it is fairly clear that the approach, although carried out here with help of pictorial forms for structures, is suitable for computer processing. We have already prepared a computer program for deriving atom codes by enumeration paths.[15] The recursive nature of the algorithm should not give an impression of

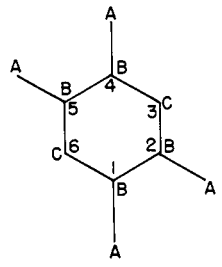

Figure 15. The 1,2,4,5-tetramethylcyclohexane skeleton.

inefficiency. In fact, by erasing a vertex and reevaluating atom codes, efficiency of the search is only improved. This is perhaps best seen by considering the average number of labels per vertex. If each vertex has *all* possible fragment labels in a vertex-to-vertex matching, we have to examine all $k!$ possibilities (k being the number of different labels). It is clear that, in such an instance, labels and atom codes have not been of help, *but* that is true only at this stage of the search. Once we have exhausted the selected vertex, the graph is reduced by erasing that vertex, a process which will eventually reduce the number of labels. Finally, in the last steps there will be a few multiple labels and we may have only single labels, which would immediately point to the fragments. For the example discussed at the beginning, we had as mean label value[18] 3.68 and the value had dropped to 3.21 after finding the first fragment. One can view the mean label values as an argument of the gamma function (factorial function), and the values $\Gamma(3.68)$ and $\Gamma(3.21)$ give an indication of the size of the problem, as discussed and illustrated earlier.[9]

Enumeration of paths for a large and complex system may be time consuming. However, here we need truncated paths, the length of which is given by the size of the fragment sought. In order to indicate the time involved, we may mention that graph K_6 (complete graph on six vertices, each vertex connected to each other) has 975 paths (of length 1 to 5) and our ALLPATHS algorithm finds them in 18.5 seconds (using an IBM 370/148 and a PL/1 program). The number of paths for the skeleton of pimarane, when we limit path length to five and six bonds respectively, is 555 and 755 paths, respectively, from which we can estimate the time required for path finding as 10 and 15 seconds, respectively. With an increase in the size of molecule, the time will proportionally increase, while increase in the size of the fragment will increase the lengths of the truncation. In practice, it may well be discovered that, regardless of the size of the fragment, codes may usefully be truncated after five or six bond lengths, which would reduce code generation at the cost that occasionally in the list of fragments one would find a graph which is nonisomorphic to the sought-for fragment. Practice will show whether such a trade-off is useful or should be avoided.

It may appear that the suggested scheme is more or less a minor modification of a brute-force node-to-node matching. In order to better illuminate differences, we will consider another fragment, the 1,2,4,5-tetramethylcyclohexane skeleton (Figure 15).

A	1,2,3,3,4,3,1
B	3,3,3,4,3,1
C	2,4,4,2,4,2

One can, after some visual inspection, see that this particular fragment is *not* contained in the graph of pimarane. In a node-to-node matching procedure, the search will proceed similarly whether the graph contains the fragment or not. Hence, it will consume about the same time and involves a similar effort. In our scheme, the search for nonexistent fragments is also, in principle, similar to the case of hidden fragments. However, when the fragment is actually not present

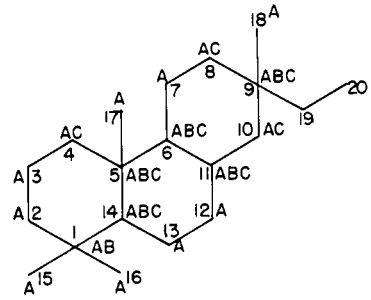

Figure 16. Graph of pimarane and multiple labeling induced by the fragment of tetramethylcyclohexane.

in the skeleton, the number of incompatible codes is likely to increase so that convergence rate of the search is increased. Figure 16 contains a multiply labeled graph of primarane where A, B, C are labels of the new fragment (tetramethylcyclohexane). Vertices 19 and 20 can be erased immediately, reducing the graph slightly. However, inspection of connectivity will show *without* full vertex-to-vertex matching that in this case no fragment F is present. First, consider label A, which in the fragment must be adjacent to B. This screening of Figure 16 immediately shows that the condition is not met for atom 3, which has no other label and can be erased. Then, atoms B in the fragment require another adjacent atom B. This condition is not met at atom 9 and, as a consequence, atoms 8 and 18 cannot have label A (the previous constraint), which eliminates atom 18 from the further analysis. Similarly, C is situated between two B atoms, but that condition is not fulfilled at atoms, 4, 8, 9, 10, and 11. The number of labels has been reduced appreciably. We could now also eliminate vertex 8, since it has "lost" both its labels. One can continue in this way examining whether all B labels *simultaneously* have adjacent labels A, B, C. This is not the case for atom 1. Therefore, by deleting B at site 1, we find that atoms 2, 15, and 16 no longer can have label A and must be erased. Once label B has been removed from atom 1, we find that atom 14 also does not fulfill the requirement (having an adjacent A, B, C simultaneously) and can no longer have the label B, which immediately eliminates atoms 1 and 13 from further consideration. Atom 14 also cannot have label C after we have removed label B from atom 1. The consequence of this is that atom 5 no longer satisfies the condition of simultaneous adjacency of A, B, C. We could continue until we finally find that all vertices are incompatible with the adjacency requirements of the fragment, but already, by deleting label B at atom 14, we are left with only *three* additional labels B. Because the fragment requires *four* B vertices, the skeleton of primarane *cannot contain the fragment of 1,2,4,5-tetramethylcyclohexane*. We arrived at this conclusion *without* atom-to-atom matching. Instead, *adjacencies* (the nearest-neighbor relationships only) were examined. This is a kind of node-to-node match, but involving only one atom at a time and independent of other matches. In usual node-to-node matching, one proceeds from first neighbors to second, third, and so on until such a mismatch occurs or full agreement is found, and backtracks at any branching point. Instead of such random node-to-node matching, by previously finding all paths (in a systematic way, and at required depth), we have been able to eliminate much of the random character of the search and at the same time increase the efficiency of the search. The scheme outlined needs further extensions to be fully adapted for searches of actual molecules (presence of multiple bonds, heteroatoms, stereospecifity, related properties). We are optimistic that such realistic applications are well within the reach and capability of the scheme and expect that applications of this approach to real problems will emerge.

ACKNOWLEDGMENT

The support of this research through National Science Foundation Grant CHE-76-21295 is gratefully acknowledged. We also acknowledge partial support from U.S. Department of Energy, Division of Basic Energy Research, administered through R. S. Hansen, a grant to Ames Laboratory, Iowa State University.

REFERENCES AND NOTES

(1) On leave from Ames Laboratory, Iowa State University, Ames, Iowa 50011.
(2) For a recent survey, see R. C. Read and D. G. Corneil, *J. Graph Theory*, **1**, 339 (1977).
(3) L. C. Ray and R. A. Kirsch, *Science*, **126**, 814 (1957).
(4) J. E. Ash, "Connection Tables and Their Role in a System", in "Chemical Information Systems", J. E. Ash and E. Hyde, Ed., Wiley, New York, 1975, p 167.
(5) J. Figueras, *J. Chem. Doc.*, **12**, 237 (1972).
(6) G. Saucier, *R. I. R. O.* 5e annee, No. R-3 (1971) 31, MR 46.1654 (as quoted in ref 2).
(7) E. H. Sussenguth, *J. Chem. Doc.*, **5**, 36 (1965).
(8) M. M. Cone, R. Venkataraghavan, and F. W. McLafferty, *J. Am. Chem. Soc.*, **99**, 7668 (1977).
(9) M. Randić, *J. Chem. Inf. Comput. Sci.*, **18**, 101 (1978).
(10) L. Euler, *Comm. Acad. Sci. Imp. Petropol.*, **8**, 128 (1736); reprinted in N. L. Biggs, E. K. Lloyd, and R. Wilson, "Graph Theory 1736-1936", Clarendon Press, Oxford, 1973, p 3.
(11) A. V. Aho, *Acta Crystallogr.*, **33**, 5 (1977). E. L. Lawler, *Math. Centre Tracts*, **81**, 3 (1976). R. E. Tarjan, "Complexity of Combinatorial Algorithms", Computer Science Dept., Stanford University, Report STAN-CS-77-606, April, 1977.
(12) M. Randić, *J. Chem. Inf. Comput. Sci.*, **17**, 171 (1977).
(13) M. Randić, *J. Chem. Soc., Faraday Trans. 2.*, submitted for publication.
(14) M. Randić, *Chem. Phys. Lett.*, **58**, 180 (1978).
(15) M. Randić, G. M. Brissey, R. B. Spencer, and C. L. Wilkins, *Comput. Chem.*, in press.
(16) M. Randić, *Chem. Phys. Lett.*, **42**, 283 (1976); *Croat. Chem. Acta*, **49**, 643 (1977); *Int. J. Quant. Chem.*, in press.
(17) M. Randić, Proceedings of Bremen Conference: "Mathematical Structures in Chemistry" (June 1978, Bremen, Germany), to appear in MATCH (Informal Communications in Mathematical Chemistry, edited by A. T. Balaban, A. S. Dreiding, A. Kerber, and O. E. Polansky).
(18) Label A has to be counted with weight 2, because it represents two atoms.

Graph Theoretical Approach to Recognition of Structural Similarity in Molecules

MILAN RANDIĆ*[1a] and CHARLES L. WILKINS*[1b]

Department of Chemistry, University of Nebraska–Lincoln, Lincoln, Nebraska 68588, and Ames Laboratory, Iowa State University, Ames, Iowa 50010

Received June 8, 1978

In many applications, one is faced with the problem of identifying similar structural forms. This is usually performed in an intuitive manner and the outcome may be ambiguous. Here a well-defined approach for determining the degree of similarity among structures (molecular skeletons and more general graphs) is suggested. We select paths in a structure as the invariants upon which comparisons among structures should be based. For each structure, one first enumerates paths of different length and constructs a sequence of path numbers for the atoms (vertices). From such a list of atom codes, one can derive, by summing the contributions of individual atoms, a sequence of path numbers for a molecule (or a graph). The comparison of structures, thus, can be transformed into a comparison of *sequences* which are suitable for rigorous mathematical analysis. It is *assumed* that similar sequences imply similar structures, and, for selected examples, the validity of this assumption has been demonstrated. Molecules, generally considered similar, have been found to have similar sequences of path numbers, and molecules differing considerably in their connectivity show large differences in their path numbers. As an illustration of the concept of similarity based on path enumerations, we consider the problem of selecting from a set of structures those most similar to naturally occurring monocyclic monoterpenes. We used available computer-generated hypothetical monocyclic monoterpenes. The motive for this comparison is the assumption that potentially interesting skeletal forms should show considerable similarity to those found in natural structures. Such a technique may be of particular use in problems in chemical taxonomy.

INTRODUCTION

In many applications, one is confronted with the task of identifying a few structurally related systems among dozens or hundreds of others which are similar. One must begin by defining the concept of *similarity* which may apply to a selected property, to a dominant property, or to the overall features of a system (or objects) under examination. We will be concerned here with the *structural* parameters of molecules, each structure being specified by molecular connectivity (i.e., the adjacency or structural matrix).[2] In order to use this definition of similarity on a quantitative level, we require a measure of similarity and specified tolerance levels which can be used to categorize structures as similar or dissimilar.[3] For example, we may wish to review physical and chemical properties of alkanes (boiling points, solubility, vapor pressure dependence on temperature, viscosity, optical density, chromatographic retention volumes, density, etc.), in particular isomeric variations. To call two isomers similar (with respect to a single selected property, e.g., boiling point), we have to specify tolerance levels and then see if differences among the isomers fall within the specified interval. Objections to such an empirical approach mainly arise from its a posteriori character, which dictates that whatever conclusion is drawn is of limited guidance in a completely new situation. For example, similarity in boiling points and/or a few other thermodynamic properties cannot assist in predicting which members of a group of therapeutically useful compounds will show the optimum antihistaminic, anticholinergic, antipsychotic, antidepressant, analgesic, or anticonvulsive potency. Here, it would be more useful to have several representative compounds of each class (standards) and then to compare the candidate structure with the standards, using as many properties as thought to be pertinent. Since many molecular properties, and especially chemical or therapeutic activity, bear some relationship to chemical structure, studies of the similarity of *structures*, rather then *properties*, should be the first

0095-2338/79/1619-0031$01.00/0