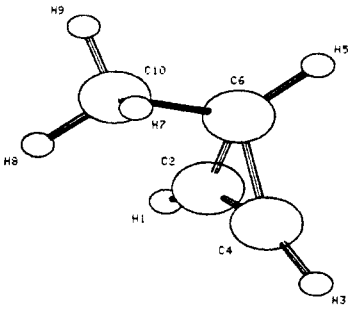

Figure 4. PLUTO plot of 3-methylcyclopropene.

After this step a reasonable geometry for most molecules is obtained. A sample printout for D2DD/M, i.e., 3-methylcyclopropene, is shown in Figure 3.

Plotting of Molecular Structures. After the geometry has been calculated, the program prepares a plotting file so that a three-dimensional ball-and-stick plot of the molecule may be obtained. The molecular plotting program used is a modified version of PLUTO, a program developed by Sam Matherwell at the University Chemical Laboratory at Cambridge, England. The PLUTO plot of 3-methylcyclopropene is shown in Figure 4.

Quantum Mechanics Program. The atomic coordinates generated by the model builder are transferred into a quantum mechanics program to obtain the molecular orbitals, charge densities, and heat of formation of the molecule. These results may then be used later for the calculation of other molecular properties. Currently the quantum mechanical method which is interfaced with the Molecular System Identification routine (MSI) is the MINDO/3 program.[9]

The program is currently executed interactively on a VAX-11 computer.[10] The complete calculation of a 20-atom molecule requires approximately 20 s of CPU time, including geometry minimization and SCF calculations.

In the succeeding papers of this series we will examine the property synthesizers, presently at various stages of development.

ACKNOWLEDGMENT

We thank the National Science Foundation (Grant No. Env 77-74061) and the National Cancer Institute (Contract No. NO1-CP-75927) for supporting our work.

REFERENCES AND NOTES

(1) S. R. Wilson and J. H. Huffman, *J. Org. Chem.*, **45**, 560 (1980).
(2) M. Razinger, J. Zupan, M. Penca, and B. Barlic, *J. Chem. Inf. Comput. Sci.*, **20**, 158 (1980).
(3) See for example (a) S. W. Benson, F. R. Cruickshank, D. M. Golden, G. R. Hangen, H. E. O'Neal, A. S. Rodgers, R. Shaw, and R. Walsh, *Chem. Rev.*, *69*, 279 (1969); (b) J. T. Chou and P. C. Jurs, *J. Chem. Inf. Comput. Sci.*, **19**, 3 (1979).
(4) (a) C. A. Shelley, H. B. Woodruff, C. R. Snelling, and M. E. Munk, *ACS Symp. Ser.* **No. 54**, 92 (1977); (b) H. B. Woodruff, C. R. Snelling, C. A. Shelley, and M. E. Munk, *Anal. Chem.*, **49**, 2075 (1977); (c) H. B. Woodruff and M. E. Munk, *J. Org. Chem.*, **42**, 1761 (1977).
(5) E. G. Smith, "The Wiswesser Line Formula Chemical Notation", McGraw-Hill, New York, 1968.
(6) See, for example, A. J. Stufer and P. C. Jurs, *J. Chem. Inf. Comput. Sci.*, **16**, 99 (1976).
(7) T. M. Dyott, A. J. Stuper, and G. S. Zander, *J. Chem. Inf. Comput. Sci.*, **20**, 28 (1980).
(8) For other model builder routines, see, for example, W. T. Wipke, S. R. Heller, R. J. Feldman, and E. Hyde in "Computer Representation and Manipulation of Chemical Information", Wiley, New York, 1974.
(9) R. C. Bingham, M. J. S. Dewar, and D. M. Lo, *J. Am. Chem. Soc.*, **91**, 1285 (1975).
(10) The program is available for use by others. For copies, contact G. Klopman.

Computer Perception of Topological Symmetry via Canonical Numbering of Atoms

MILAN RANDIĆ,*[1] GREGORY M. BRISSEY, and CHARLES L. WILKINS*

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

Received August 15, 1980

A previously described algorithm for perception of topological symmetry has been programmed for computer use. The algorithm is based on the concept of the smallest binary code for a graph and requires generation of all numberings for a structure which will produce the canonical form for the adjacency matrix. The unique adjacency matrix corresponds to the smallest possible (binary) number representing the structure when its rows are read from top to bottom and from left to right. Operation of the program is illustrated with a selection of polycyclic structures. Typically, molecules with a dozen carbon atoms and several rings produce results in a few hundred tests as compared with $N!$ which would be considered in an exhaustive procedure.

INTRODUCTION

Although the problem of recognition of symmetry for rigid bodies and rigid molecular frames is straightforward, the same task in the case of nonrigid structures and particularly for graphs in which only connectivity is specified is much more difficult. Even the case of constitutional symmetry for polycyclic structures projected in two dimensions may pose difficulties, since oblique projections frequently obscure equivalence of sites. In many problems, in particular in computer manipulations with structures, it is of interest to establish the symmetry of the structure and the equivalence of atoms in the structure. The problem has received attention in recent literature, and several alternative schemes have been advocated.[2] The problem, however, which is closely related to the problem of graph isomorphism, ordering of graphs, and graph construction, is sufficiently involved that it appears desirable to continue to pursue the subject and report on alternative schemes. One justification for such continuing interest is the lack of clear-cut evaluations of one method vs. another and the lack of knowledge about how close existing schemes approach an idealized optimal algorithm. Furthermore, it is entirely possible that certain approaches may be superior to others for various applications. Therefore we report here another approach to the determination of constitutional symmetry (or atom equivalence) in molecules. Our method is quite general and, in fact, is a means of determining the symmetry

0095-2338/81/1621-0052$01.25/0

of graphs. The problems of constitutional molecular symmetry, the symmetry of nonrigid structures, and the symmetry of structures of transient chemical reaction intermediates are only special cases of the problem of symmetry of graphs. Molecules, in contrast to graphs, may have various kinds of atoms and different types of bonds. Such additional features only simplify the determination of the symmetry, as they impose additional constraints which must be satisfied. Hence, by addressing the more difficult problem of determining symmetry of graphs, the simpler subset of chemical applications will also be solved.

The question of graph symmetry arises in various problems of physics and chemistry besides those already mentioned. In problems of statistical mechanics, as discussed in the 1940s by Mayer and Mayer,[3] determination of graph symmetry proved helpful in reduction of certain definite integrals. For enumeration of isomers, the symmetry of related graphs is of interest in construction of the cycle index for the Polya counting theorem.[4] In the study of chemical transformations, in particular rearrangements, symmetry is of interest, as it may lead to simpler pictorial representations for the processes.[5] Symmetry of graphs is also of interest in certain problems of algebra of coupling of vectors, such as in Wigner $3n-j$ symbols, and their pictorial representation.[6] Graphs appearing in the various kinds of problems mentioned may be widely different in size and complexity. Molecular graphs are relatively simple, from the point of view of symmetry, in that it is rare that all or many atoms belong to the same equivalence class. On the other hand, graphs associated with rearrangements typically have all vertices equivalent, i.e., would be described in mathematical literature as transitive[7] (more precisely, vertex transitive). The approach that we will outline is general and applies equally to both types of problems. It will be seen that transitive graphs may require considerable memory for storing all alternatives to be further screened in the process. Thus, some modification of the existing program, if it is to be applied to complex graphs normally found in studies of isomerizations, is desirable. Such a modification has been developed and applied in discussions of the symmetry of complex chemical graphs,[8] but it has not been implemented in the present program. Therefore, the present program, although general enough to be used for any situation, may not be the most practical approach for graphs of the complexity mentioned above, and we do not intend here to advocate indiscriminate use. Constitutional symmetry of molecules, as already mentioned, does not involve the problems typical for transitive graphs, i.e., the presence of an excessive number of initial combinatorial possibilities, and the present approach is very suitable for this particular application.

In this paper we will outline the scheme and the algorithm and will proceed by discussing a number of illustrations, with all the computational details. These demonstrate the practicality of the method and at the same time provide enough information for comparison with other methods. Direct comparison with other currently available programs will not be made, primarily because previous outlines and illustrations used by other authors do not report sufficient details to make such comparisons meaningful. We consider it particularly desirable to report on the number of trials that the computer makes in arriving at the final answer, which can be an indication of the number of steps required to get the result.

OUTLINE OF THE METHOD

It is known that the problem of graph symmetry (i.e., the automorphism problem) is computationally equivalent to the problem of graph identification (i.e., the isomorphism problem).[9] Both problems are closely related to ordering of graphs[10] and graph construction. Hence a similar scheme with minor modifications and modified emphasis can serve all the four related needs. Examples of such related applications of canonical numbering of atoms have been reported.[11,12] Past applications of the particular canonical numbering for determining symmetry of graphs have been confined to transitive graphs (Petersen graph,[13] Desargues–Levi graph,[14] Balaban's graph for the homotetrahedryl rearrangement,[8] and some equally complex graphs representing rearrangements of other molecular species[15]). A brief outline of the method also has been reported,[16] so it suffices here to emphasize those computational aspects which indicate the efficiency of the program and serve for comparison of the computations between different molecules and different methods.

Before comparing different methods, it is useful to recognize the underlying principles of techniques being examined. All approaches are directed toward the goal of discriminating among all atoms (i.e., atomic environments). If it is found that two atoms have the same characterization, they are candidates for equivalence. If different, clearly the atoms are not equivalent. The uncertainties and difficulties lie with the lack of assurance that a selected set of graph invariants provide a *complete* characterization. In contrast, canonical labels are assigned uniquely, and cases of multiple choice correspond to the presence of symmetry. The difficulty here is not in uncertainty but in developing a *practical* scheme, a scheme which can arrive at the desired labeling without screening an excessive number of combinatorial possibilities.

Clearly, in general, this is an $N!$ problem, and the question can be raised whether it is possible, even in principle, to circumvent the $N!$ problem. In the language of the complexity theory,[17] the question is whether the graph isomorphism problem can be reduced to one solvable by a polynomial-type algorithm. The prevailing view is that this will not be possible. Nevertheless, in chemical applications "the worst case" situation is infrequent. Thus, problems of symmetry of chemical constitutional forms may be more tractible even though the algorithm used may formally qualify as exponential. We hope to demonstrate that this is indeed the case.

Any *rule* for labeling atoms in a molecule, which leads to a unique assignment of labels (or unique up to the automorphism), can qualify for the task outlined above and can be called canonical. For a rule to be practical it must be possible to arrive at the canonical labeling without screening an excessive number of alternatives. We have previously shown that one can efficiently find a numbering of atoms in a molecule which results in an adjacency matrix which, when its rows are read from left to right and from top to bottom, corresponds to the smallest binary representation. Thus, this approach appears practical. One cannot exclude the possibility of a structure which will elude such a *practical* solution, but accumulated experience suggests that molecular graphs are not complex enough to cause many problems.

In application, one starts with an *unlabeled* graph and then assigns the smallest label 1 to all qualified vertices. At this stage, all vertices of the lowest valency qualify, since they would introduce in the first row of the adjacency matrix the smallest number of nonzero entries, which clearly is necessary for the row to correspond to the smallest possible binary number. This step is illustrated on a simple graph in Figure 1, which offers two possibilities. Next, we must assign the largest available labels to the neighbors of atom 1, which are labels 4, 5, because then and only then the nonzero entries will be preceded by as many zeros as possible, resulting in the smallest number: 0 0 0 1 1. When this is done, the four alternatives of the second row of Figure 1 result. Now we have to look for the assignment of the next smallest label, 2. The remaining unassigned place then belongs to label 3. The four possibilities now produce for the neighbors of label 2 either 3, 5 or 3, 4, with the corresponding second row being 0 0 1

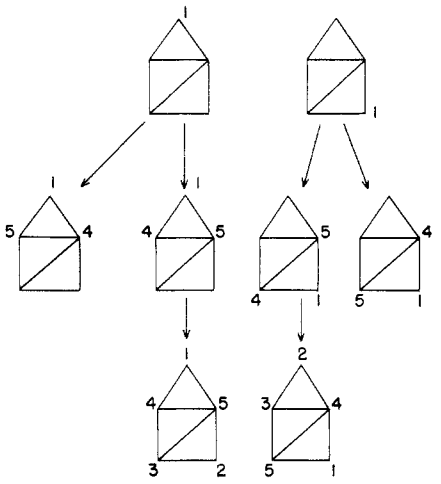

Figure 1. Illustration of the gradual assignment of labels in the search for canonical labeling.

0 1 and 0 0 1 1 0, respectively, since we will assign 2 to the site with only two neighbors, rather than three.

Clearly the first choice, the neighbors 3, 5, corresponds to a smaller binary code and is accepted. To complete the search for the canonical labeling we must *test* the resulting labelings for the labels 3, 4, 5 not yet considered in order to be sure that they are associated with the smallest possible binary codes for the corresponding rows. In the example of Figure 1 both alternatives lead to the same adjacency relations: 3 has the neighbors 2, 4, 5; 4 has the neighbors, 1, 3, 5, and 5 has the neighbors 1, 2, 3, 4 in *both* alternatives. Therefore, *both* labelings qualify and the structure has equivalent atoms and nontrivial symmetry. To identify the equivalent atoms, we list the atoms of one structure and below each label indicate the label in the other structure:

1	2	3	4	5
2	1	4	3	5

Thus, the pair 1, 2 and the pair 3, 4 are equivalent, while atom 5 is unique. The associated symmetry operations can be derived by listing the permutations (in the above case there is only one, besides trivial identity) of labels between the first and any subsequent acceptable labeling.

ALGORITHM AND PROGRAM

In order to implement this approach for computer processing, we must resolve the question of how to handle unlabeled graphs. Typically, structures are recorded in computer-readable form either as a list of neighbors or as an adjacency matrix. These use some arbitrary labeling. It is possible, at least in the case of acyclic graphs, to list sequences of paths *without* specifying connectivity[18] and thus contain in a computer memory all essential information on an acyclic graph without the use of labels. However, such input requires one to reconstruct the graph, and that problem may be quite difficult, even if the concept of unique characterization of graphs by path sequences can be extended to polycyclic structures. We resolved the problem of reconciling the need for labels and the use of unlabeled graphs by adopting arbitrary labels but assigning to each vertex in the process of bookkeeping zero as the label. Having all labels equal is equivalent to having no labels. The search for canonical labeling is now initiated, and vertices are assigned labels 1 to n. The process ends when no zero labels remain and all acceptable solutions are tested for canonicity.

The flow chart for the program as implemented in FORTRAN and BASIC is shown in Figure 2. One starts with an arbitrary labeled graph and determines the effective valency of the

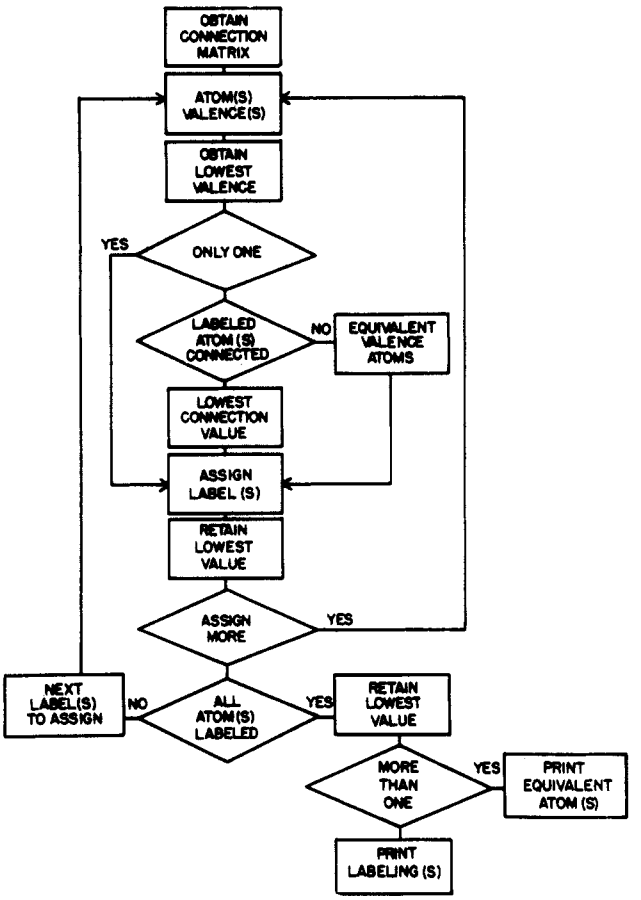

Figure 2. Flow chart for the program which produces the canonical labeling and the list of equivalent vertices.

atoms. The effective valency is given by the valency of the atom in the molecular graph (i.e., by the number of connected atoms) less the number of already labeled connected atoms. Therefore, the effective valency indicates the number of *new* labels that are required for each atom at a particular stage of the canonical assignment. Only the atoms with the smallest effective valencies are considered at each stage.

An effective valency of zero indicates an atom all of whose neighbors have already been assigned; a valency of one indicates an atom with a single unlabeled neighbor, etc. If there is only one atom of the lowest effective valency, one continues by assigning the smallest available label to this atom and returns to determine the next atom having the smallest effective valency. If several atoms have the smallest effective valency, these are checked for existing labeled vertices, and those associated with the smallest labels (no additional vertices, or vertices with the largest labels already assigned) are chosen for further consideration. If, as a result, only one atom qualifies, it is assigned the smallest label, and its neighbors receive the next largest available labels in all possible combinations, the number of which is given by $v_e'!$, where v_e' indicates the effective valency.

In the case of organic molecules, $v_e'!$ is 1, 2, and 6, at worst, in the situation where the carbon atom has four neighbors, three of which are not assigned.[19] The labelings are tested at each step for the binary value for the corresponding row of the adjacency matrix, and only those which give the smallest binary numbers are kept. The same applies if there is more than a single vertex of the same smallest effective valency—this situation only leads to a larger number of possibilities to be tested, and if these satisfy the minimal label condition, they are kept for further processing.

Incomplete assignments, which are represented in the computer output as a list of labels in which zeros are present, are

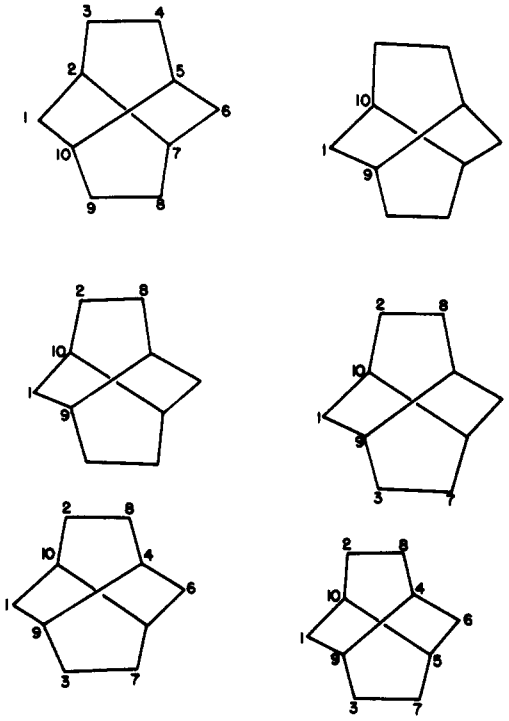

Figure 3. Twistane skeleton and illustration of a few steps in the search for canonical labelings. Each partially labeled graph corresponds to the first possibility as shown in Table I for partial assignments when a new label is introduced. Zeros in Table I correspond to unlabeled vertices.

kept, and at each successive step reexamined. They either generate additional partially labeled graphs or eliminate some of the possibilities of the previous step which no longer qualify because they lead to binary values for the next row which are larger than alternatives. The process continues until all labels are used.

Next, one tests the resulting numberings for the rows not directly constructed to ensure that they correspond to the smallest binary values. As output one obtains a list of equivalent vertices, new canonical labels, and the adjacency matrix in canonical form. As an option, one can print the intermediate steps of partial assignment, which illustrates the gradual assignment of labels. Each possibility examined is *counted* as a test but is recorded only if at that particular step in the process it qualifies as the smallest binary value for the row (label) considered. Hence, the number of tests actually performed is larger than the number of registered partial assignments. The efficiency of the search should be judged by the number of tests made. Therefore the output includes the number of tests. This number should be always compared to $N!$, the number of all possibilities that an unorganized exhaustive labeling scheme would require in order to assure one of having all possible canonical labelings, i.e., the list of equivalent vertices.

In Table I we illustrate the use of the program on twistane, $C_{10}H_{16}$, an isomer of adamantane (Figure 3), which has been arbitrary labeled. It should be mentioned that the arbitrarily selected labeling adopted for storing the connectivity of the molecular graph does not affect the rate of the search process, because the computer assumes unlabeled vertices, and labels only serve as references. The output lines of Table I are grouped in sections, each section being associated with consideration of a *new* label, only the smallest labels being counted. Thus the first 12 lines signify 12 alternative ways of assigning label 1 to the 6 atoms of valency 2, each such possibility further producing 2 alternative ways of assigning labels 10 and 9 to its neighbors. As an indication of the efficiency of the algorithm it should be noted that this is a 10! problem (over 3.5×10^6 possibilities).

Table I. Illustration of the Input and Output for the Case of Twistane (Figure 3)

Input (Connectivities)		Output (Adjacency Matrix)									
1 with	2, 10	0	1	0	0	0	0	0	0	0	1
2	1, 3, 7	1	0	1	0	0	0	1	0	0	1
3	2, 4	0	1	0	1	0	0	0	0	0	0
4	3, 5	0	0	1	0	1	0	0	0	0	0
5	4, 6, 10	0	0	0	1	0	1	0	0	0	1
6	5, 7	0	0	0	0	1	0	1	0	0	0
7	2, 6, 8	0	1	0	0	0	1	0	1	0	0
8	7, 9	0	0	0	0	0	0	1	0	1	0
9	8, 10	0	0	0	0	0	0	0	1	0	1
10	1, 5, 9	1	0	0	0	1	0	0	0	1	0
Partial Assignments for Label 1: 12 Possibilities											
1	10	0	0	0	0	0	0	0	0	9	
1	9	0	0	0	0	0	0	0	0	10	
0	10	1	9	0	0	0	0	0	0	0	
0	9	1	10	0	0	0	0	0	0	0	
0	0	10	1	9	0	0	0	0	0	0	
0	0	9	1	10	0	0	0	0	0	0	
0	0	0	0	10	1	9	0	0	0	0	
0	0	0	0	9	1	10	0	0	0	0	
0	0	0	0	0	0	10	1	9	0	0	
0	0	0	0	0	0	9	1	10	0	0	
0	0	0	0	0	0	0	10	1	9	0	
0	0	0	0	0	0	0	9	1	10	0	
Partial Assignments for Label 2: 8 Possibilities											
1	10	2	8	0	0	0	0	0	9		
1	9	0	0	0	0	0	8	2	10		
2	10	1	9	0	0	0	0	0	8		
0	0	9	1	10	2	8	0	0	0		
0	0	8	2	10	1	9	0	0	0		
0	0	0	0	9	1	10	2	8	0		
0	0	0	0	8	2	10	1	9	0		
2	8	0	0	0	0	0	9	1	10		
Partial Assignments for Label 3: 4 Possibilities											
1	10	2	8	0	0	0	7	3	9		
1	9	3	7	0	0	0	8	2	10		
0	0	8	2	10	1	9	3	7	0		
0	0	7	3	9	1	10	2	8	0		
Partial Assignments for Label 4: 4 Possibilities											
1	10	2	8	4	6	0	7	3	9		
1	9	3	7	0	6	4	8	2	10		
6	4	8	2	10	1	9	3	7	0		
6	0	7	3	9	1	10	2	8	4		
Partial Assignments for Label 5: 4 Possibilities[b]											
1	10	2	8	4	6	5	7	3	9		
1	9	3	7	5	6	4	8	2	10		
6	4	8	2	10	1	9	3	7	5		
6	5	7	3	9	1	10	2	8	4		

[a] The initial adjacency matrix is included in the output to make sure that no error was made on input. The partial assignments provide an optional output which illustrate the stepwise assignment of labels. Each line represents one viable (at that point) partial assignment. At each successive step another (small) label is added. The first line for each successive step has been illustrated in Figure 3. The number of possibilities tested in order to arrive at those shown as viable partial assignments may be larger and is given at the end of the output (Table II). [b] This completes the assignment of all labels. The resulting final four possibilities are now further checked to see if they correspond to the smallest binary codes for the labels 6–10. (For the above example this is the case; hence the last partial assignments provide final acceptable labelings.)

The efficiency can be appreciated by examining the number of *unproductive* combinations not even considered. The number of possibilities of combining 3 labels (such as 1 with 10 and 9) from 10 is given by the binomial coefficient $\binom{10}{3}$ which is 120, so already at the first step 119 alternatives are eliminated (each of which would generate 12 possibilities since

Table II. Essential Part of the Output: Correspondence between the Initial (Arbitrary) Labels and the Derived Canonical Labels,[a] The Canonical Form for the Adjacency Matrix, and the List of Equivalent Atoms[b]

	Initial (Arbitrary) Labels									
	1	2	3	4	5	6	7	8	3	10
	Canonical Labels									
A	1	10	2	8	4	6	5	7	3	9
B	1	9	3	7	5	6	4	8	2	10
C	6	4	8	2	10	1	9	3	7	5
D	6	5	7	3	9	1	10	2	8	4

Canonical Form for the Adjacency Matrix

0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	1	1	0	0	1
0	0	0	1	1	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0

Equivalent atoms:	Initial Labels	Canonical Labels
	1, 6	1, 6
	2, 5, 7, 9	4, 5, 9, 10
	3, 4, 8, 9	2, 3, 7, 8

CPU time: 0.47 s; total number of tests: 40 out of 10!

[a] In this case the four possibilities shown as A-D. [b] The CPU time (for an IBM 370/158) and the total number of tests made are given.

there are 6 atoms, each with 2 neighbors, with the smallest valency). From this total of 1440 possibilities, only 12 are listed as qualified.

As is seen from Table I most vertices are unassigned at this stage, but the assignment of label 1 to atoms having three carbon atoms as neighbors has been discarded. In the next step, label 2 is considered, and all possibilities having the effective valency of 1 (i.e. requiring only one new largest label) qualify for examination. However not all offer optimal labeling. For the second row to correspond to the smallest binary code, label 2 should be adjacent to label 10, not 9. This eliminates some of the previous possibilities and results, in all, in eight viable alternatives of partially assigned graphs to be further studied. As can be seen from Table I, introduction of label 3 further reduces the possibilities to only four, which remain valid after successive assignments of labels 4 and 5 and the test for canonicity involving the remaining labels. The output, in Table II, lists both the old (arbitrary) labels and the new labels satisfying the rules. The canonical form for the adjacency matrix is given, as is the total number of tests, which is 40 in this example. Thus, in only 40 tests we have succeeded in locating the four canonical labelings out of a few million possibilities!

Although this does not prove that each application will be similarly efficient, it is an indication of the properties of the algorithm, suggesting its use in other situations. In order to obtain a proper assessment of the method and algorithm, it

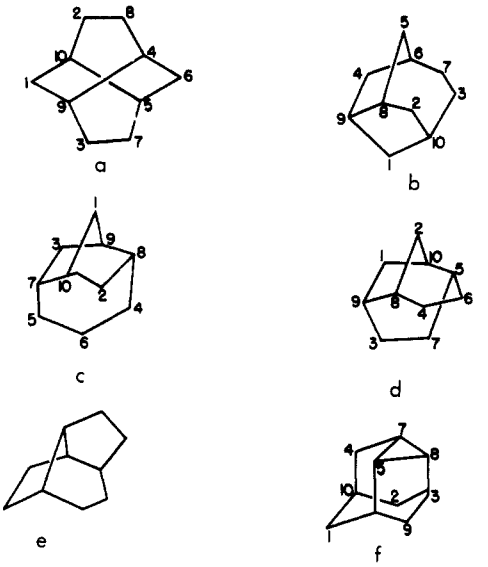

Figure 4. Structures whose results are summarized in Table III.

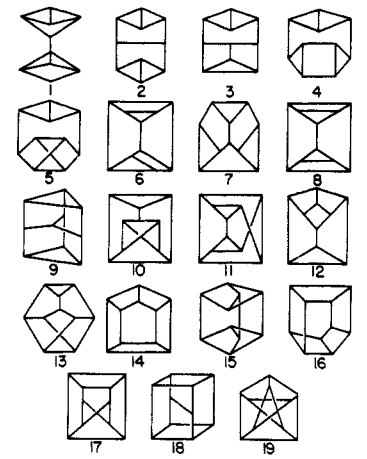

Figure 5. The 19 trivalent graphs of 10 vertices for which details of the search for canonical numberings are given in Table IV. The canonical labeling can be found in ref 11.

seemed useful to apply the approach to a selection of representative molecular structures. Accordingly we have done so. Such illustrations suggest the domain of applicability, the complexity of problems, the computer needs, and other factors of interest.

ILLUSTRATIONS

In selecting examples we decided to consider groups of related molecules, as a comparison between them may highlight inadequacies of the algorithm, and this procedure reduces subjectivity. Table III summarizes the results for the ten-carbon polycyclic systems (which, with one exception, are tetracyclic) of Figure 4. Computation times in seconds are

Table III. Summary of Canonical Numbering for 10-Carbon Polycyclic Structures[a]

structure	time, s	partial assignments	order	equivalent atoms	no. of tests
a	0.47	12, 8, 4, 4, 4	4	(1,6) (2,3,7,8) (4,5,9,10)	40
b	0.57	12, 16, 18, 8, 4	4	(1,2,4,5) (3,7) (6,10) (8,9)	72
c	0.57	12, 10, 10, 10, 10	2	(1) (2,3) (4,5) (6) (7,8) (9,10)	64
d	0.46	12, 8, 4, 2, 2	2	(1,2) (3,4) (5) (6,7) (8,9) (10)	44
e	0.52	12, 4, 16, 4	2	(1,3) (2,8) (4,5) (6,10) (7,9)	84
f	0.75	8, 10, 6, 2, 2	2	(2,6) (3,5) (7,8) (1) (4) (9) (10)	38

[a] Time in s of CPU use, the partition of viable partial labelings showing how each new label influences the number of alternatives to be tested, the order of the associated symmetry group (i.e., the number of acceptable labelings), and the number of tests made are shown (graphs appear in Figure 4).

Table IV. Summary of the Search for Canonical Numbering in Trivalent Graphs of $n = 10$ Vertices[a]

structure	time, s	partial assignment	order	equivalent atoms	no. of tests
1	1.99	60, 24, 16, 32	32	(1,3,8,10) (2,4,7,9) (5,6)	212
2	1.34	60, 8, 16, 16	16	(1,3,6,8) (2,7) (4,5,9,10)	212
3	1.34	60, 20, 2, 4	4	(1,4) (2,3) (5,7) (8,10)	164
4	1.85	60, 12, 32, 32	8	1(4) (2,3) (5,10) (6,7,8,9)	216
5	1.55	60, 24, 8, 16	16	(1,4) (2,3) (5,10) (6,7,8,9)	184
6	1.39	60, 16, 8, 4	2	(1,7) (2,8) (3,10) (4,9) (5,6)	172
7	5.25	60, 168, 48, 48	6	(1,3,4,5,9,10) (2,6,8)	516
8	1.30	60, 24, 8, 4	4	(1,5,6,10) (2,4,7,9) (3,8)	172
9	6.94	60, 192, 72, 72	12	(1,2,3,7,8,9) (4,6,10)	564
10	1.52	60, 12, 24, 8	4	(1,3) (4,10) (5,9) (6,8)	196
11	0.85	60, 8, 8, 8	8	(1,5,6,10) (2,4,7,9) (3,8)	196
12	1.42	60, 24, 6, 12	6	(1,3,7) (2,4,6) (8,9,10)	186
13	1.35	60, 16, 4, 2	2	(1,3) (4,10) (5,7) (8,9)	188
14	1.34	60, 40, 20, 20	20	(1,2,3,4,5,6,7,8,9,10)	180
15	2.89	60, 24, 48, 48, 48	48	(1,3,5,6,8,10) (2,4,7,9)	300
16	1.58	60, 40, 20, 20, 20	20	(1,2,3,4,5,6,7,8,9,10)	200
17	1.50	60, 24, 4, 4	4	(1,2) (3,4,9,10) (5,6,7,8)	156
18	1.52	60, 16, 16, 16	8	(1,3,8,10) (2,4,7,9) (5,6)	204
19	6.42	60, 240, 120, 120	120	(1,2,3,4,5,6,7,8,9,10)	660

[a] Clearly, the first partial assignments always lead to 60 possibilities, since label 1 can take any of ten sites, and with each one may associate labels 10, 9, 6 in 3! ways (graphs appear in Figure 5).

Table V. Summary of the Search for Canonical Labeling and Atom Equivalence for Selected 14-Carbon Polycyclic Structures[a]

structure	time, s	partial assignments	order	equivalent atoms	no. of tests
A	0.73	12, 4, 4, 24, 8, 8	2	(1,4) (2,5) (3,7) (6,11) (8,12) (10,14)	112
B	0.74	12, 4, 4, 24, 8, 4	2	same as above	112
C	0.58	12, 6, 2, 12, 4, 2	1	all different	72
D	2.63	12, 12, 12, 72, 72, 72	4	(1,2,5,6) (3,4) (7,8,12,13) (9,11) (10,14)	324
E	0.95	12, 10, 6, 36, 24, 8	2	(1,2) (3) (4) (5,7) (6,8) (9,10) (11) (12,13) (14)	136
F	0.84	12, 8, 6, 36, 12, 4	1	all different	136
G	0.80	12, 8, 4, 24, 16, 8	4	(1,4) (2,3,5,7) (6,8,11,12) (9,10,13,14)	96

[a] CPU times in s (graphs appear in Figure 6).

Table VI. Selection of Diverse Polycyclic Structures and the Numbering of Vertices Which Required from a Fraction of a Second to a Few Seconds[a]

structure	time, s	partial assignment	order	equivalent atoms	no. of tests
H	0.93	48, 16, 4	4	(1,3,5,7) (2,6) (4,8)	132
I	0.58	8, 32, 16	8	(1,2,5,6) (3,4,7,8)	72
J	2.57	12, 12, 12, 72, 72, 72	12	(1,3,5,8,10,12) (2,4,6,7,9,11)	252
K	0.67	6, 24, 12, 12, 12	6	(1,4,11) (2,3,5,7,8,10)	78
L	0.92	16, 14, 6, 2, 2, 1	1	none	96
M	0.56	16, 14, 6, 4, 2, 2, 2, 1	1	none	69
N	1.26	14, 14, 12, 4, 4, 4, 4, 4, 4	4	(1,11,15,17) 1 (2,6,12,14) (3,5) (7,13) (9,18) (10,16)	112
O	3.93	6, 2, 8, 24, 48, 48, 48, 12, 16, 8, 4, 2, 2, 2	2	(27,28)	304
P	1.44	22, 10, 4, 1, 1, 1, 1, 1	1	none	182

[a] Graphs appear in Figure 7.

also shown (IBM 370/158) and are well below 1 s. The number of viable partial assignments in each case is shown as a sequence of numbers. For example, in the case of twistane (a) it is 12, 8, 4, 4, 4. The next line contains the number of acceptable solutions (i.e., the order of the corresponding symmetry group) and is followed by a list of equivalent vertices in their canonical form, and finally the number of tests made is indicated. All structures in Figure 4 required fewer than 100 tests. The graphs in Figure 5 are trivalent regular graphs with 10 vertices and are more complex than the molecular graphs of Figure 4. Now all vertices at the initial stage qualify for label 1 (all having the same valency), and the number of edges is increased. Nevertheless, as we see from Table IV, the required computer time has barely doubled and the number of tests remains quite low, usually below 200.

More complicated cases we have examined include the results summarized in Table V for the polycyclic structures having 14 carbon atoms shown in Figure 6. The spread of computer time required by the individual structures of Figure 6 varies considerably, from about 0.5 s to over 2.5 s, and is approximately paralleled by the variations in the number of tests performed. This number should be compared with 14! which is approximately 8.7×10^{10}. Observe also in Table V the sequences indicating the bookkeeping for partially assigned graphs. These sequences may expand after some initial contraction, as several examples show to be the case with the assignment of label 4. In one of the structures all subsequently generated possibilities remain "active" until the final test for the canonicity selects the four uniquely describing the structure.

Finally in Table VI are tabulated results for a number of odd polycyclic structures (Figure 7) to show that apparently complex skeletal forms need not require more time and more tests than seemingly simple structures. For instance, 8-carbon tricyclooctane takes approximately the same time as the pentacyclic 16-carbon structure, which even requires fewer tests! The 12-carbon "iceane" (a structure having bonds in a arrangement found for hydrogen bonds in ice) appears to be the structure requiring the most time among those considered. In part, the increased time is associated with the

Table VII. A Few Graphs Requiring Bookkeeping for a Large Number of Combinatorial Possibilities Illustrating the Times Required and the Number of Tests Performed[a]

structure	time, s	partial assignment	order	equivalent atoms	no. of tests
Q	16.03	108, 432, 84, 16, 16, 8, 8	4	(1,5,12,16) (2,4,13,15) (3,14) (6,8,9,11) (7,10) (17,18)	1292
R	36.44	120, 480, 180, 60, 60, 120, 60, 60	20	(1,2,3,4,5,6,7,8,9,10) (11,17,18,19,20) (12,13,14,15,16)	1800
S	12.31	216, 288, 288	84	(1,2,3,4,5,6,7,8,9)	936
T	1.18	12, 4, 32, 32, 32, 32	4	(1,3) (2,4) (5,7) (8,12) (9,11)	152
U	6.94	20, 8, 4, 8, 8, 8, 80, 128, 96, 64, 32	32	(1,3,5,7) (2,6) (4,8) (11,17) (13.15) (12,16,19,20)	576
V	1.38	24, 24, 24, 8, 8, 16, 16, 16	16	(1,2,4,5,10,11,15,16) (3,6,9,12) (7,8,13,14)	176
W	1.66	72, 16, 16, 32	16	(1,6,7,12) (2,5,8,11) (3,4,9,10)	232
X	3.61	108, 24, 4, 4, 16, 16, 32	32	(1,8) (2,3,6,7) (4,5) (11,18) (12,13,16,17) (14,15)	412
Y	13.98	408, 24, 8, 2, 2, 2	1	none	1580

[a] The dramatic efficiency of the algorithm proposed is even better illustrated for some of these complex graphs. Although the number of vertices are doubled in comparison with graphs of Tables III–VII, the number of tests increased at most by an order of magnitude, in contrast to the enormous increase in going from $N!$ to $(2N)!$ (graphs appear in Figure 7).

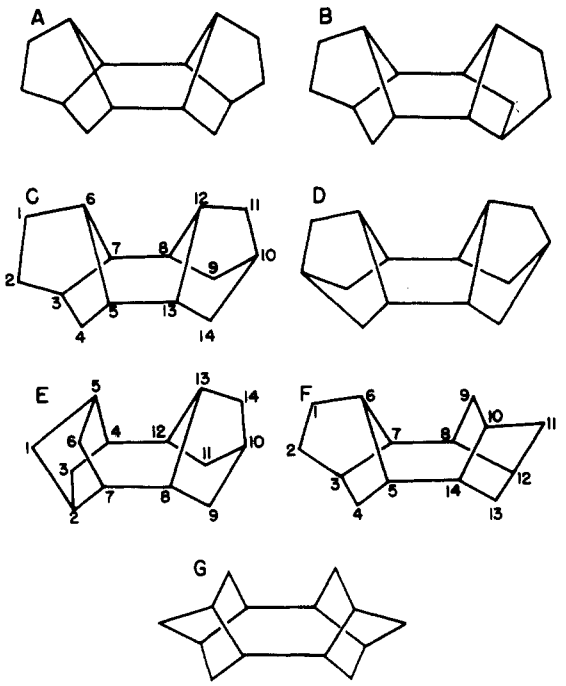

Figure 6. Selected pentacyclic structures $C_{14}H_{20}$ summarized in Table V.

higher symmetry of such structures, so it is not that unproductive searches cause more time but that the particular structures have a larger number of acceptable solutions.

The program has also been tested on a few structures of relatively high symmetry in order to indicate the limitations of *practicality* of the approach. Furthermore, we considered a number of structures that have appeared as illustrations in papers by other authors. In Table VII we summarize the results for some graphs from the mathematical literature, such as Blanusa's graph (derived by fusion of two Petersen's graphs)[20] and Isaac's graph (a member of the elusive "snarks")[21,22] as well as a simplex polycorypha bounded by six pentahedra (a graph of some interest in chemistry for relating the conformers of propane, $CH_3CH_2CH_3$, when the methyls are freely rotated). These graphs require an order of magnitude more time and an order of magnitude more tests. The graphs of Blanusa and Isaac contain 18 and 20 vertices, respectively, which is the major cause for the increased work on the problems, while the 9-vertex simplex has increased valency; hence $v_e'!$ generates a larger number of permutations to be examined for each given labeling. Currently the program limits the number of viable possibilities tested at each step to 500, which will exclude some highly symmetrical graphs from consideration. Time will also be a factor in consideration of larger molecular graphs. However, the number of atoms, when these are of variable valence, need not be small. This is illustrated in the case of one of the sterols in Table VI, the testing of which involved only 316 attempts and took less than 4 s.

CONCLUSION

The algorithm outlined appears to be a practical way of determining equivalence of atoms as well as isomorphism of structures. As illustrated with selected examples, the application appears to extend to generally more complex graphs than required for manipulation of chemical structures. The operational practicality will depend on the routine requirements and frequency of use, since occasional more difficult and time-consuming problems, such as listing the symmetry operations for a dodecahedron or a four-dimensional cube, will usually be considered only once. The increase in time and memory requirements primarily reflects the increased solutions in such situations and not reduced efficiency of the algorithm. One useful aspect of the program is the possibility of listing partially labeled graphs at any intermediate step of the process. If a particular problem appears intractible and prevents further execution due to lack of available memory, one can stop and initiate the problem with partially assigned labels, examining selected possibilities only.

The framework of the approach permits further refinements for the study of high symmetrical graphs of regular graphs of higher valency. These are the graphs which will initially produce a great number of partially assigned possibilities to be further tested, and their number may exceed available memory capacity on small computers. However, the approach in its present form can be applied to some graphs of moderate complexity which have high symmetry or high valency and are regular or transitive. This has been illustrated with the case of the "snark" of Table VII (which is a vertex-transitive, regular graph of valency 3 having 20 vertices) and on the graph with 17 vertices (Y) (which is regular of valency 4 but is not vertex transitive and, in fact, has no equivalent vertices). This particular graph has been found by Mackey[24] to lead to local minima in a search for canonical labels which is constrained to pairwise permutations of labels—an algorithm initially suggested[25] as an *alternative* to the search for canonical labels. As shown here, the concept of canonical labels based on the minimal binary code applies to isomorphism and automorphism problems, the valid algorithm being one using unlabeled

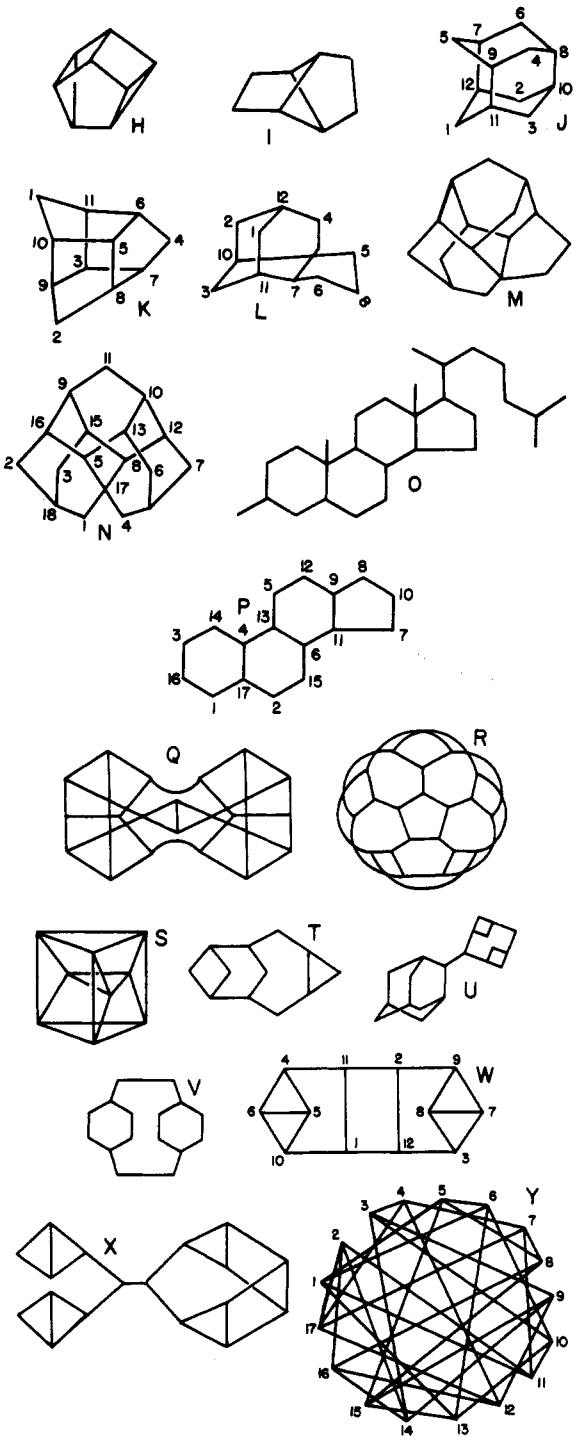

Figure 7. Additional structures for which the canonical labeling procedure was tested.

graphs[11,25] as implemented in the program outlined in this paper. One may wonder why these somewhat unusual graphs, such as Mackay's 17-vertex construction, or graphs from the mathematical literature[26] should concern us here, where our prime interest is in chemical applications. Besides several examples of such graphs that arise in chemical applications (suggested mostly in works of Balaban[27]), use of such complex graphs is likely to reveal more clearly the limitations and differences between alternative approaches, and we believe such graphs should be considered before finally evaluating the merits of various approaches.

ACKNOWLEDGMENT

The support of the National Science Foundation through Grant CHE-79-10263 is gratefully acknowledged.

REFERENCES AND NOTES

(1) On leave from Ames Laboratory, Iowa State University, Ames, IA 50011.
(2) A selection of more recent publications is given in ref 3.
(3) C. A. Shelly and M. E. Munk, *J. Chem. Inf. Comput. Sci.*, **17**, 110 (1977); R. E. Carhart, *ibid.*, **18**, 108 (1978); C. Jochum and J. Gasteiger, *ibid.*, **17**, 113 (1977); M. Uchino, *ibid.*, **20**, 116 (1980); **20**, 121, 124 (1980).
(4) J. E. Mayer and M. G. Mayer, "Statistical Mechanics", Wiley, New York, 1940.
(5) G. Polya, *Acta Math.*, **68**, 145 (1937); J. D. Dunitz and V. Prelog, *Angew. Chem.*, **80**, 700 (1968); K. E. De Bruin, K. Naumann, G. Zon, and K. Mislow, *J. Am. Chem. Soc.*, **91**, 7031 (1969); A. T. Balaban, *Rev. Roum. Chim.*, **18**, 841 (1973).
(6) R. J. Ord-Smith, *Phys. Rev.*, **94**, 1227 (1954); B. R. Judd, "Operators Techniques in Atomic Spectroscopy", McGraw-Hill, New York, 1963; A. Jucys, J. Levinsonas, and V. Vanagas, "Matematiceskij Apparat Teorii Momenta Kolicestvos Dvizenia", Lietuvos TRR, Vilni'us, 1960, pp 83–114.
(7) F. Harary, "Graph Theory", Addison-Wesley, Reading, MA, 1969.
(8) M. Randić, *Int. J. Quant. Chem.*; *Quant. Chem. Symp.*, in press.
(9) For a bibliography on graph isomorphism see R. C. Read and D. G. Corneil, *J. Graph Theory*, **1**, 339 (1977); G. Gati, *J. Graph Theory*, **3**, 95 (1979); C. J. Colbourn, Technical Report No. 123/78, Department of Computer Science, University of Toronto, 1978; B. Weisfeiler, "On Construction and Identification of Graphs" Springer: Berlin, 1976.
(10) J. F. Nagle, *J. Math. Phys. (N.Y.)*, **7**, 1588 (1966).
(11) M. Randić, *J. Chem. Inf. Comput. Sci.*, **17**, 171 (1977).
(12) M. Randić, *Acta Crystallogr., Sect. A*, **A34**, 275 (1978).
(13) M. Randić, *Croat. Chem. Acta*, **49**, 643 (1977).
(14) M. Randić, *Int. J. Quant. Chem.*, **15**, 663 (1979).
(15) M. Randić and V. Katovic, to be published.
(16) M. Randić, *Chem. Phys. Lett.*, **42**, 283 (1976).
(17) R. E. Tarjan, "Complexity of Combinatorial Algorithms", Technical Report, Computer Science Department, Stanford University, 1977; E. L. Lawler, *Math. Centre Tracts*, **81**, 3 (1976); A. V. Aho, *Acta Crystallogr.*, **33**, 5 (1977); R. Bellman, K. L. Cooke, and J. A. Lockett, "Algorithms, Graphs and Computers", Academic Press, New York, 1970.
(18) M. Randić, *J. Chem. Inf. Comput. Sci.*, **18**, 101, 1978; M. Randić, *MATCH* **7**, 5 (1979).
(19) Although carbon atoms have a maximal valency of four in molecular graphs, there will always be some carbons of smaller valency which will be assigned before; hence eventually the effective valency of quaternary carbon will be reduced to three or less. Maximal valency of four is possible in mathematical graphs, such as the simplex based on six pentahedra, discussed later in the text.
(20) D. Blanusa, *Glas. Mat-Fiz-Astron., Ser. II*, **1**, 31 (1945).
(21) R. Isaacs, *Am. Math. Monthly*, *82*, 221 (1975).
(22) For a readable account of problems associated with ref 19 and 20 see: M. Gardner, *Sci. Am.*, *234*, No. 4, 126 (1976).
(23) M. Randić, unpublished.
(24) A. L. Mackay, *J. Chem. Phys.*, **62**, 309 (1975).
(25) M. Randić, *J. Chem. Phys.*, **60**, 3920 (1975).
(26) A good source on complex graphs can be found in H. S. M. Coxeter, "Regular Polytopes", Dover Publications, New York, 1973; H. S. M. Coxeter, "Regular Complex Polytopes", Cambridge University Press, New York, 1974.
(27) A. T. Balaban, *Rev. Roum. Math. Pures Appl.*, **17**, 3 (1972); **18**, 1033 (1973); *Rev. Roum. Chim.*, **18**, 841 (1973); **22**, 243 (1977); A. T. Balaban and F. Kerek, *ibid.*, **19**, 631 (1974); M. Gielen, R. Willem, and J. Brocas, *Bull. Soc. Chim. Belg.*, **82**, 617 (1973); M. Gielen and J. Topart, *J. Organomet. Chem.*, **18**, 7 (1969); M. Gielen, *Med. Vlaam. Chem. Ver.*, **31**, 201 (1969); J. F. Arens and J. Roy, *Neth. Chem. Soc.*, **94**, 3 (1975).