Detection of Constitutionally Equivalent Sites from a Connection Table

Bo Tao Fan, Alain Barbu, Annick Panaye, and Jean-Pierre Doucet*

Institut de Topologie et de Dynamique des Systèmes, CNRS URA 34, Université Paris 7,
1 rue Guy de la Brosse, 75005 Paris, France

Received November 13, 1995[⊗]

A simple and efficient algorithm for the perception of constitutionally equivalent atoms in a target molecule is reported. The algorithm consists of three steps with input data from a connection table. Our algorithm was tested on a number of typical complex structures. Results show that, with this algorithm, detection of topologically equivalent sites can be performed rapidly, and a solution (which up to now seems to be complete) can be obtained. Performance and computation time are also discussed.

INTRODUCTION

During the development of our research program on spectral simulation and structural elucidation, several new algorithms related to molecular structure handling are proposed, including the detection of constitutionally equivalent sites.

Partitioning of atoms for a target molecule into classes of equivalent sites constitutes an essential treatment in computer handling of chemical structures. This problem intervenes for instance in the search for a canonical description of molecules or in some spectral analyses, as for example in ^{13}C or ^{1}H NMR, to specify the number of peaks.

Two distinct approaches of detecting topological equivalence of sites (often referred as "topological symmetry") appeared:

−An always exact solution, mathematically established, would result from a systematic search for local graph automorphism. But this "brute force approach" implies large computer tasks to CPU time and storage requirements.

−For practical applications various methods, generally relying on the concept of extended connectivity, appeared as faster and less demanding on computer resources. These methods, however, must be considered as giving only necessary (but not automatically sufficient) conditions and must be better regarded as screening techniques allowing for reducing the number of (time-consuming) automorphism searches. If two sites appeared as nonequivalent at any level, they are surely nonequivalent. If after examination with all proposed tests they are still considered as equivalent, a complete automorphism search must be carried out. Although nonrigorously established, this detection of equivalent sites must be thought as largely time-saving in limiting the number of necessary rigorous comparisons for survival sites.

As already said, this latter approach generally relies on the extended connectivity concept and the well-known Morgan algorithm.[1] Advantages of these methods are speed and simplicity, but they fail to give the real automorphism partitioning for many structures, for instance, molecule no. 1, shown in Figure 1, as quoted by Munk *et al.*[2] Some modified Morgan's algorithms, known as "growing a spanning tree" method, were later developed by Wipke,[3] Corneil and Gotlieb,[4] and Munk.[2] Perception of equivalent sites with these algorithms takes into account the necessary atomic properties to produce complete partitioning.[2] These works largely improved the development in this field.

In order to solve problems regarding unique and unambiguous coding, automorphism partitioning and computation of symmetry groups, a number of other algorithms were later developed.[5−13] Gasteiger and Randić used a technique based on a canonical numbering of atoms;[7,8] Munk proposed an algorithm using vector-classification and a permutation construction procedure;[9] Uchino reported a vector function method which is based on manipulation of invariant vectors and bond matrices.[10] Work reported by Rucker concerns the perception of topological symmetry by mathematically manipulating the connectivity matrices.[11] Recently Varmuza *et al.* developed an algorithm within the framework of graph theory by a spectral substructure search in automorphism mapping.[12]

These algorithms were implemented in different chemical computer programs, but were generally speaking faced to four types of problems:

−Many approaches are not complete solution.[14]

−Some algorithms require an important storage space and are very time-consuming. For example, for structure no. 17 in Figure 3, the required storage space in Gasteiger's algorithm is $2 \times (6 \times 36 + (10 + 17) \times 17 + (2 + 17) \times 17 + 50) = 2.1$ Ko, according to the formula given by author.[7] Randić reported,[8] for this structure, a calculation time of 14 s on an IBM 370/158 machine.

−Computation complexity is also a problem. The complexity of Munk's algorithm[2] depends upon the numbers of classes and that of vectors in each class (complexity $= \Pi_i(C_i)!$). For instance, the complexity for the relatively simple structure no. 13 (in Figure 3) will be $8!4!4! = 23\ 224\ 320$.

−Some algorithms need many laborious manipulations of matrices, which imply both important computation time and large storage space.

In the current development of our spectral simulation and structural elucidation programs, within framework of DARC system, the connection table serves as basic input or output data. We therefore developed a new algorithm for the perception of constitutionally equivalent atoms from a connection table. This new approach falls therefore into that category of "a priori approximate" methods relying on the extended connectivity concept. Our concern was the search for an efficient approach, limiting as far as possible the need for brute and comprehensive automorphism searches.

[⊗] Abstract published in *Advance ACS Abstracts,* June 15, 1996.

S0095-2338(95)00171-5 CCC: $12.00

ALGORITHM

For any molecular structure, the connection table contains all the structural information (nature of atoms, connectivity degree, bond types, stereochemistry if an extended connection table is used, etc...). The information about constitutionally equivalent atoms should be also included in this table. In chemistry, it is known that the term "two constitutionally equivalent atoms" implies the following common characteristics:

(1) They are atoms of same nature (atomic number).

(2) They have the same neighboring atoms on *N* depth levels.

(3) They possess the same bond types on *N* depth levels.

These properties make it possible to use, for the detection and extraction of constitutionally equivalent atoms, a description of the atom local environment by means of a subspanning tree (SST): a description neighbor to the concept of FREL (Fragment Reduced to Environment that is Limited) largely used as a basic concept in the DARC system.[15]

Starting from a root atom (the atom under scrutiny, called "focus" in a FREL), these SST are built up in concentric levels in a breadth first process.

In fact two types of SST will be used: a Maximum Overlapping Spanning Tree (MOST) involving for any level all the atoms present at a given topological distance. A MInimum, Not overlapping, Spanning Tree (MINST) defined with the constraint that, when an atom is used in a layer, it cannot be present again in the next layers. This constraint is very useful, since it makes it possible to avoid the problems due to presence of ring systems. Figure 1 gives an example of those trees.

Before coming into more details to the algorithm, we define two different SST: similar SST and identical SST.

Similar SST: If two SST have for every layer the same composition, it means same members, same corresponding atom natures, and same corresponding bond types then these two SST are similar.

Identical SST: If two **similar SST** possess the following property: for each corresponding layer, the atoms involved are either the same, or equivalent sites in two SST, they are identical SST. In other words, if two sites are candidate for equivalence, their neighbors must also be equivalent.

With these definitions, we can conclude that two equivalent atoms must have identical SST on N depth levels. This is a necessary and sufficient condition.

Let us suppose that S_1 and S_2 are two equivalent sites, T_1 and T_2 their corresponding SST of N depth levels, and A_i (a_{i1}, a_{i2}, ..., a_{in}) and A'_i (a'_{i1}, a'_{i2}, ..., a'_{in}) the atom vectors involved at layer *i*. Supposing that layers *i* are not identical for two SST, it means that we cannot find for at least one atom of this layer in T_1 its equivalent atom in T_2, for example "Not $a_{ij} \leftrightarrow a'_{ij}$". Because all atoms of layer i are derived from layer *i*-1, the fact "Not $a_{ij} \leftrightarrow a'_{ij}$" implies that we can find at layer *i*-1 for T_1 and T_2 at least two atoms with different environment. These two atoms are therefore not equivalent. The same deduction can be applied to *i*-2, *i*-3, ... layers. Finally, the root atoms (focus) are not equivalent, so S_1 and S_2 are not equivalent. This result is contradictory with the hypothesis. We thus conclude that if S_1 and S_2 are two equivalent sites, T_1 and T_2 must be identical.

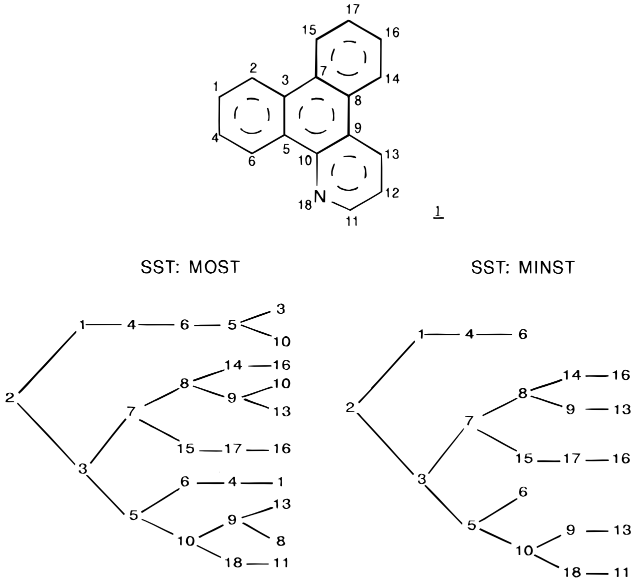


Figure 1. Example of two types of FRELS (MOST and MINST) for atom no. 2.

If T_1 and T_2 are identical, atoms involved at each layer of two SSTs are either the same or equivalent, including the layer directly derived from the focus. It means that for this layer of two SSTs, they must have the same environment, including the preceding layer, the focus. Hence S_1 and S_2 must be equivalent.

In fact, common Morgan type algorithms used up to now for the problem of topological symmetry tackle the problem of constitutionally equivalent atoms at the level of similar SSTs (MOSTs) but not topologically identical SSTs (MINSTs). Taking into account, as example, structure n°1 (Figure 1), the MOSTs centered on atom 2 and atom 14 covering the whole molecule, are illustrated in Chart 1.

In these MOSTs, all atoms are carbon, except atom 18. But this atom unfortunately appears in two MOSTs at the

Chart 1

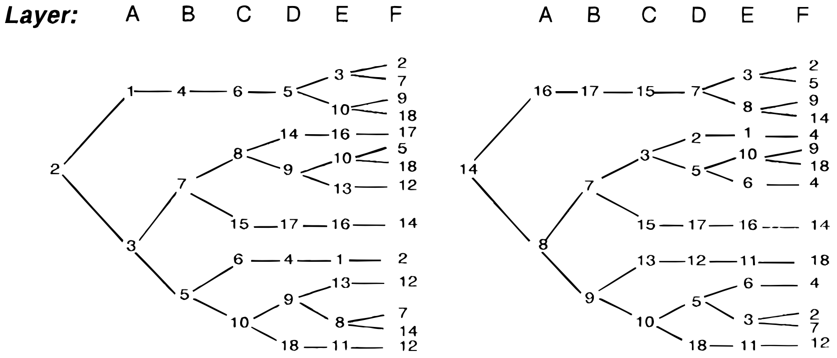


Table 1. Parameter Values (Structure No. 16 in Figure 3)

no.	initial value	final value	no.	initial value	final value
1	6462	466 649 220	11	6963	976 391 328
2	6963	597 718 989	12	6963	786 899 958
3	6462	535 311 864	13	6462	535 311 864
4	6963	786 899 958	14	6963	597 718 989
5	6462	535 311 864	15	6462	535 311 864
6	6963	597 718 989	16	6462	466 649 220
7	6462	535 311 864	17	6963	597 718 989
8	6462	535 311 864	18	6462	535 311 864
9	6963	786 899 958	19	6462	535 311 864
10	6963	976 391 328	20	6963	786 899 958

same layers. The two MOSTs are therefore similar. Common Morgan-based algorithms would therefore classify these two atoms as equivalent.[2] However, because of the presence of nitrogen atom 18, all atoms obviously are not equivalent in this molecule.

Our algorithm consists of three main parts:

−Atoms are first classified by means of a characteristic identifier value based on MOST.

−Within these classes, possible "identical MINSTs" are sought for.

−Due to the approximate character of the first classification, the preceding step must be iterated to refine the detection of the classes of equivalence.

PART I: CATEGORIZATION−SEARCHING FOR SIMILAR MOST

The following steps are sequentially carried out:

Assignment of Initial Identifier Values for each Atom from a Connection Table. Three parameters are used to assign an initial identifier value (V): nature of atoms (A), degree of connectivity (D), and bond type (B) (1: simple bond; 3: conjugated bond; 4: double bond; 6: triple bond). We have

$$V_i = 1010A_i + 100D_i^2 + \sum_j (\nu_i)_j (B_i)_j^2$$

D^2 and $(B_i)^2$ is preferred in order to increase the difference of values between two atoms, due to their different degrees of connectivity. $(\nu_i)_j$ represents the number of same bond-types stemming from atom i to its neighbors j. An example is shown in Table 1 (column "initial value") for structure no. 16 in Figure 3.

Updating the Identifier Values. Updating identifier values is carried out for each atom by adding to the initial atom value the identifier values of its neighbors. The initial value is then replaced by this sum. This process is performed in an iterative manner and progressively carried out over all atoms depending on their distance to the focus (the atom under scrutiny).

After the first iteration, we obtain for a given atom i

$$V_i = 1010\sum_{j=i}^{P} A_j + 100\sum_{j=i}^{P} D_j^2 + \sum_{m=i}^{P}\sum_{j} (\nu_m)_j (B_m)_j^2$$

where P means all neigboring atoms. With k iterations, the result will be

$$V_i = 1010\sum_{j=i}^{P_k} A_j + 100\sum_{j=i}^{P_k} D_j^2 + \sum_{m=i}^{P_k}\sum_{j} (\nu_m)_j (B_m)_j^2$$

P_k means all neigboring atoms included in a distance of k. In the worst case where the molecule is a linear aliphatic chain, it can be shown that after k iterations the identifier value for atom i is

$$V_i = \sum_{m=0}^{2k} a_m (1010A_{i-k+m} + 100D_{i-k+m}^2 + \sum_j (\nu_{i-k+m})_j (B_{i-k+m})_j^2)$$

The coefficient a_m can be calculated as following manner: for

$$a_0 = 1;\ a_1 = a_{0(k-1\text{th iteration})} + a_{1(k-1\text{th iteration})}$$

and

$$a_n = \sum_{l=0}^{2} (a_{n-1})_{(k-1\text{th iteration})}$$

with the first values (for $k = 1$): $a_0 = 1$, $a_1 = 1$, and $a_2 = 1$.

If the aliphatic molecule consists of n atoms, and atom i is the central one, $i = n/2$ or $i = n/2 + 1$ when n is even, and $i = (n-1)/2$ when n is odd. We consider here only the case $i = n/2$ (n even), because in the computer the type "integer" (in the "C" language) gives the same result for $n/2$ and $(n-1)/2$ when n is odd. The indices ($i-k$ when $m = 0$) and ($i + k$ when $m = 2k$) become ($n/2-k$) and ($n/2+k$). This updating of identifier values must be carried out by going over the whole molecule. It means, for two stemming directions of central atom i, $(n/2-k) = 1$ and $(n/2+k) = n$. We need therefore $k = n/2 - 1$ or $k = n/2$ (if we take central atom $i = n/2 + 1$ for n even, the same results will be obtained). Since for two equivalent sites (with same final identifier values), a supplementary addition should not modify the equality of these values, we take then $k = n/2$ for the number of iterations in our algorithm.

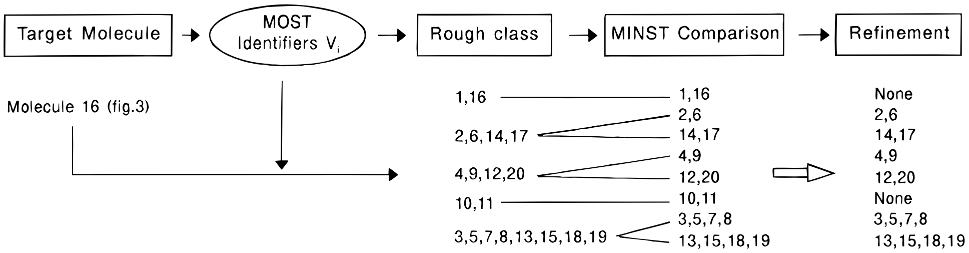


Figure 2. Procedure of searching for equivalent sites.

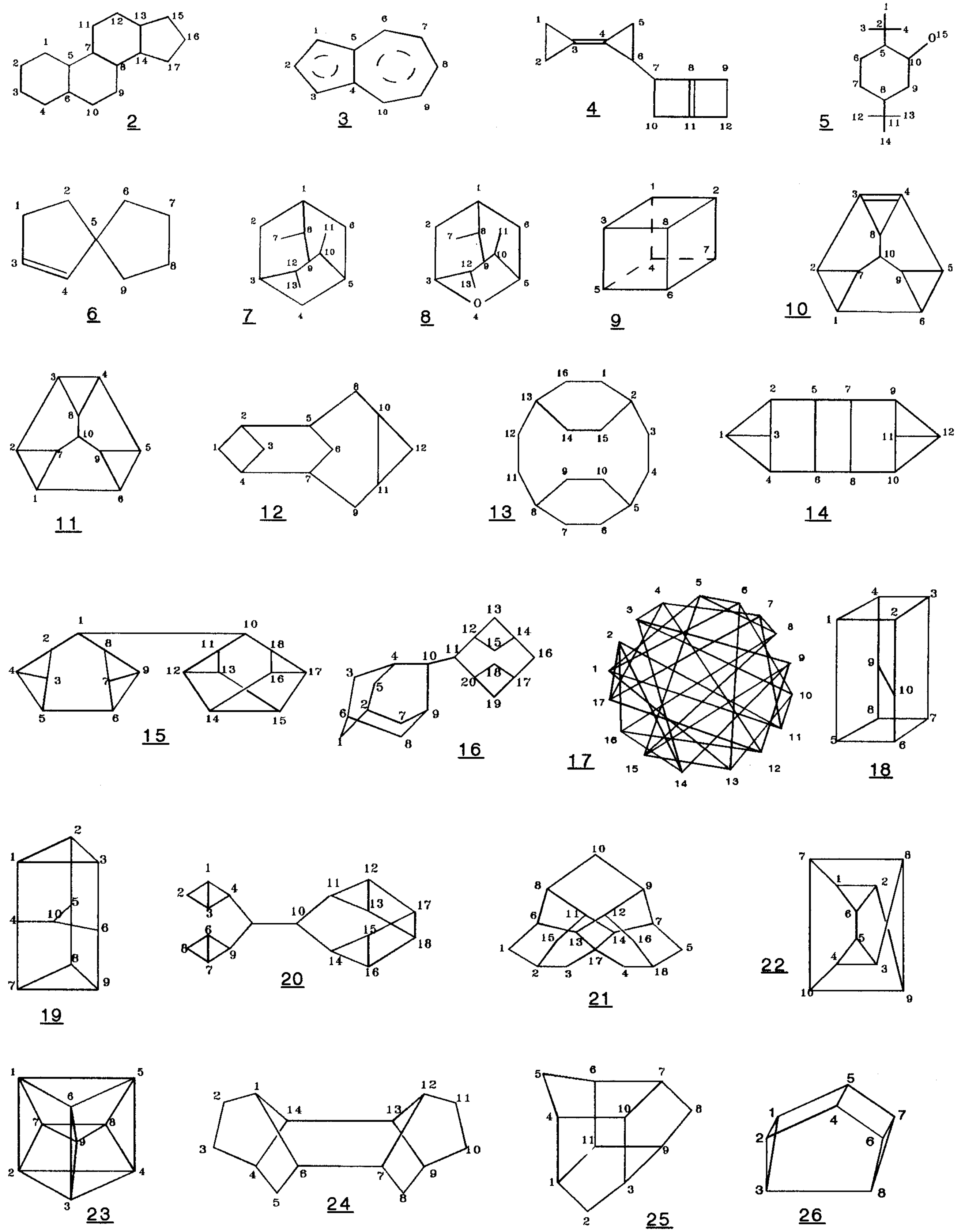
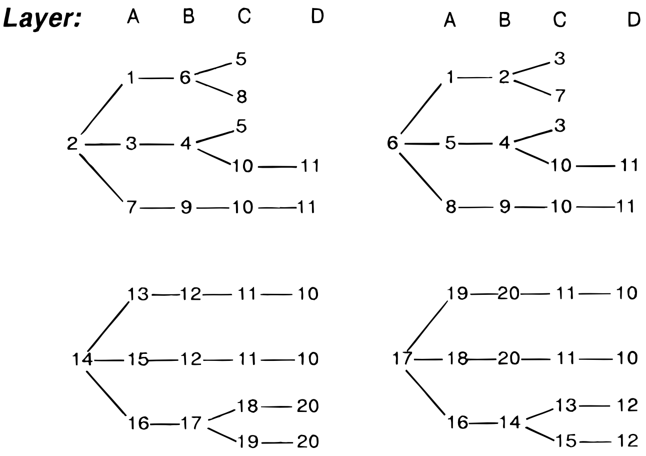


Figure 3. Examples of the tests of algorithm.

After this step, the final identifier values are obtained. In Table 1, these final values are given in row “final value” for the structure no. 16 of Figure 3.

Comparison of Final Identifier Values to Get a Rough Classification of Equivalent Sites. Until now, in spite of several modifications, the steps described above are essentially similar to Morgan type algorithms. So, the comparison of these modified identifier values only gives an incomplete solution, for instance, for the structure discussed (structure n°16 in Figure 3), we obtain as vectors

Chart 2



of constitutionally equivalent atoms: (1,16), (2,6,14,17), (4,9,12,20), (10,11), and (3,5,7,8,13,15,18,19).

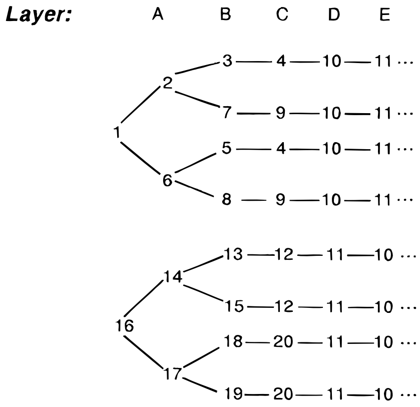
It is remarked that some of these "equivalence vectors" are not expected such as (1,16) and (10,11), and some vectors of equivalent sites are only partially correct. For example, the vector (3,5,7,8,13,15,18,19) gathers in fact two different vectors of equivalent atoms.

PART II: DETECTION OF POSSIBLE IDENTICAL MINST

To eliminate the erroneous vectors of equivalent sites and identify the different classes, we now search for identical MINST. We take as an example the vector (2,6,14,17).

At this step, we introduce the constraint that any atom already used at a level cannot intervene in further levels, and we will so consider only minimal SST. For the atoms listed in vector (2,6,14,17), we have the MINST shown in Chart 2. We begin with examination of layer A for atoms 2 and 6. Atom 1 appears in both SST. Atoms 5,7 and 3,8 are found in a same set of equivalent sites coming from the preceding categorization (3,5,7,8,13,15,18,19). The A layer for the two SST are therefore identical. The same process is applied to layer B, layer C, and so on. Finally the two MINSTs are verified as identical: atom 2 and atom 6 can be considered, at that point, as equivalent. The same comparison shows that atoms 2 and 14 do not have identical MINSTs. At layer A, atom 16 (of the MINST of atom 14) corresponds to atom 1 in the MINST of 2, since these two atoms appear in the same set of equivalent sites. Similarly atoms 13 and 15 correspond to atoms 5 and 7: the A layers are identical for the two MINSTs. The same comparison shows that layers B are also identical. But at level C the difference is evident. Layer C for the MINST of atom 14 is four-membered, while the MINST of atom 2 is five-membered. Further comparisons for next layers are then not necessary. Atoms 2 and 14 are not equivalent. Furthermore, on account of the transitivity of the equivalence relationship, atom 6 and atom 14 are therefore nonequivalent, and the comparison for these two atoms can be omitted. This first test allows us to refine the rough categorization proposed in the first part. So, class (2,6,14,17) is now separated in two parts (2,6) and (14,17) (reference Figure 3, no. 16). Finally, at that point, we obtain the following vectors of constitutionally equivalent sites: (1,16), (2,6), (14,17), (3,5,7,8), (13,15,18,19), (4,9), (12,20), and (10,11).

Chart 3



However, this step is based on the rough classification derived from the preceding step of identifiers and cannot be considered as definitive: The process fails with the vector (1,16), and (10,11). For these pairs of atoms the MINSTs (illustrated below for atoms 1 and 16) are considered as identical. This is due to the fact that actually atoms 2,6,14,17 are present in the same vector, atoms 3,5,7,8,13,15,18,19 are in another same vector, and the same situation stands for 4,9,12,20 and 10,11.

At this step (after the first test), sites detected as nonequivalent are surely not equivalent. But it may be checked that sites still considered as equivalent are really so. A further step is therefore necessary.

PART III: REFINING THE PARTITIONING

After this first test, the initial classification has to be modified, and we have to take into account these changes running the preceding test procedure once more (second test).

Taking as example the vectors (2,6) and (1,16), we can observe that this second test does not modify the result for vector (2,6). But when we return to the MINSTs of atoms 1 and 16, layers A appear now as nonidentical, because atoms 2,6,14 and 17 were separated by the first test into two different vectors.

With two tests, we find now as new classes of equivalence: (2,6), (14,17), (3,5,7,8), (13,15,18,19), (4,9), and (12,20).

A third application of the test would not modify the partitioning, and so the algorithm can stop here. We have finally detected the unique and unambiguous partitioning of constitutionally equivalent sites.

A number of molecules were used to test the algorithm (see below). In all the cases investigated we find that two test processes are sufficient for eliminating the erroneous vectors and reclassifying the vectors of equivalent sites into correct partitioning. Otherwise, the test process can be carried out several times until self-consistency is achieved (no more change appearing), and the process can be stopped.

The procedure of our algorithm for finding equivalent sites is illustrated in Figure 2.

EXAMPLES AND PERFORMANCE

According to literature results, it appears that Morgan type algorithms are generally correct for aliphatic compounds but encounter many difficulties for cyclic structures. The present algorithm was then tested with a number of aliphatic and cyclic compounds, including all model molecules investi-

Table 2. Examples and Performance in Searching for the Constitutionally Equivalent Sites[a]

no.	vectors of equivalent sites	time (s)	t_R (s)
1	none	<0.05	
2	none	<0.05	1.44
3	(1,3)(4,5)(6,10)(7,9)	<0.05	
4	(1,2)	<0.05	
5	(1,3,4)(12,13,14)	<0.05	
6	(6,9)(7,8)	<0.05	
7	(1,3,5)(2,4,6)(7,11,13)(8,10,12)	0.055	
8	(2,6)(3,5)(10,12)(11,13)	0.055	
9	(1,2,3,4,5,6,7,8)	0.055	
10	(1,6)(2,5)(3,4)(7,9)	<0.05	
11	(1,2,3,4,5,6)(7,8,9)	0.055	
12	(1,3)(2,4)(8,9)(5,7)(10,11)	0.055	1.18
13	(1,6,7,9,10,14,15,16)(3,4,11,12)(2,5,8,13)	0.165	
14	(1,3,11,12)(2,4,9,10)(5,6,7,8)	0.110	1.66
15	(2,8)(3,4,7,9)(5,6)(11,18)(12,13,16,17)(14,15)	0.275	
16	(2,6)(14,17)(3,5,7,8)(13,15,18,19)(4,9)(12,20)	0.165	6.94
17	none	0.440	13.98
18	(1,3,5,7)(2,4,6,8)(9,10)	0.110	1.52
19	(1,2,3,7,8,9)(4,5,6)	0.055	6.94
20	(1,3,6,7)(2,8)(4,9)(11,14)(12,13,15,16)(17,18)	0.110	3.61
21	(1,5,15,16)(2,18)(3,4)(8,9)(6,7,11,12)(13,14)	0.165	1.26
22	(1,2,3,4)(5,6)(7,8,9,10)	0.055	0.85
23	(1,2,3,4,5,6,7,8,9)	0.055	12.31
24	(1,12)(2,11)(3,10)(4,9) (5,8)(6,7)(13,14)	0.055	0.73
25	(1,3,4,6,7,9)(2,5,8)(10,11)	<0.05	0.67
26	(1,2,6,7)(3,8)(4,5)	0.055	0.93

[a] Time: on PC-2DX 486 machine (66 Mhz) with our algorithm; t_R: times reported by Randić on IBM 370/158 machine with their algorithm, see ref 8.

gated by Munk, Randić, and Uchino. This set of examples particularly includes structures 10−17 (Figure 3 and Table 2) where, as pointed out by Uchino,[10] Morgan-Corneil-Gotlieb-type procedures fail to give the correct automorphism partitioning. For these all compounds our algorithm gives the correct solution. For these structures (and some other examples) results are gathered in Table 2.

Because there is not any matrix manipulation, the computation is largely less time-consuming than some other methods, and the storage space is also largely reduced. For example, the computation time for structure 17 in Figure 3, is only 0.440 s on a PC-486DX2 machine. This computation time seems nearly negligible with respect to the 14 s (on an IBM370/158 machine) reported by Randić.[8] The calculation times for example molecules are given in Table 2.

Our method appears also to be significantly faster than that presented by Varmuza *et al.* in the XIth Conference on Computers in Chemical Research and Education,[12] for instance, 0.824 s against 18 s on a PC-386 (40MHz) for 3,3-di-*tert*-butyl-2,2,4,4-tetramethylpentane.

As suggested by a reviewer of the present manuscript, we have tested our algorithm with Mathon's graph A_{25},[16] used by Rucker *et al.* for the same purpose.[11b] A good result was obtained with a CPU time 3.6 s on PC-2DX 486. Rucker *et al.* have reported 11.87 s on an IBM 3090 machine.[11b]

When we carry out the comparison of MINSTs in the test steps, we need not really store whole SSTs in memory, because the comparison is performed layer by layer. Only two data vectors ("short integer" in C language) are necessary for storing actual layers in order to extract next layers. Two other data vectors ("short integers") are located in memory space (with dynamic allocation). The necessary storage space is then reduced, and it does not increase with the numbers of atoms and bonds.

CONCLUSION

A new algorithm is proposed for the perception of constitutionally equivalent atoms in a target molecules (topological symmetry), and its efficiency carefully checked on typical examples of varied complexity. Since no matrix manipulation is invoked, the procedure is fast with low storage requirement. The algorithm working from a connection table which can be directly obtained from our graphics acquisition routine offers a very simple and efficient tool in the handling of chemical structures.

REFERENCES AND NOTES

(1) Morgan, H. L. *J. Chem. Doc.* **1965**, *5*, 107.

(2) Shelley, C. A.; Munk, M. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 110.

(3) Wipke, W. T.; Dyott, T. M. *J. Am. Chem. Soc.* **1974**, *96*, 4834.

(4) Corneil, D. G.; Gotlieb, C. C. *J. ACM.* **1970**, *17*,51.

(5) Masinter, L. M.; Srindharan, N. S.; Carhart, R. E.; Smith, D. H. *J. Am. Chem. Soc.* **1974**, *96*, 7714.

(6) Golender, V. E.; Drboglav, V. V.; Rosenblit, A. B. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 196.

(7) Jochum, C.; Gasteiger, J. *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 113.

(8) Randić, M.; Brissey, G. M.; Wilkins, C. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 52.

(9) Shelley, C. A.; Munk, M. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 247.

(10) Uchino, M. *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 116

(11) (a) Rucker, G.; Rucker, C. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 187. (b) Rucker, G.; Rucker, C. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 123.

(12) (a) Varmuza, K.; Scsibrany, H. Constitutionally equivalent atoms and bonds in a molecule: a complete but simple and fast solution. First European Conference on Computational Chemistry, May 23−27, 1994, Nancy, France. (b) Varmuza, K.; Scsibrany, H. Determination of the automorphism group of graphs by a substructure search approach. ICCRE, July 17−21, 1995, Paris, France.

(13) Hu, C. Y.; Xu, L. *Chem. J. Chin. Univ.* **1994**, *15* (10), 1446.

(14) Carhart, R. E. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 108.

(15) Dubois, J. E.; Panaye, A.; Attias, R. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 74.

(16) Mathon, R *Proc. 9th S.-E. Conf. Combinatorics, Graph Theory and Computing*, **1978**, 449.

CI9501711