Ring Perception. A New Algorithm for Directly Finding the Smallest Set of Smallest Rings from a Connection Table

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

Institut de Topologie et de Dynamique des Systèmes, Associé au CNRS,
URA-34, Université Paris VII, Paris, France

Received April 20, 1993

A new simple algorithm for ring perception is reported. It directly finds the smallest set of smallest rings (SSSR) from a minimum set of data, a connection table without any classification of rings and accessory extraction procedure. Its application for some complex ring systems is presented.

INTRODUCTION

The enumeration of rings in a given structure is essential for any chemical computer system. A number of algorithms for ring perception have been reported and implemented in different computer programs. Indeed ring perception is required for identifying rings as screening substructures in database retrieval, for generating subgoals in computer-assisted synthesis, or for providing structural fragments associated with characteristic spectral features.

A comparative review was published by Lynch *et al.*,[1] who classify these algorithms in four categories: the perception of *all cycles*, the perception for a *set of simple cycles*, the perception of a *fundamental basis set of rings*, and the perception of the *smallest set of smallest rings* (*SSSR*) (reference to Figure 2).

During the current development of spectral simulation programs within the framework of the DARC system, we devised the module FINDCYC, a new algorithm for ring perception. This routine has the goal of detection and search for the smallest set of smallest rings (SSSR) in a target molecule. But it can also be used to find all cycles with a simple modification.

Several techniques have been used for finding the SSSR. Corey and Petersson[2] have developed an algorithm for use in the LHASA system. Their algorithm initially finds the set of fundamental basis rings, from which an attempt is made to derive the corresponding reduced basis, a minimum spanning set, the SSSR.

Wipke and Dyott[3] combined the Welch algorithm[4] and Gibbs algorithm[5] to find all cycles and then derived a minimum covering set, identical to a SSSR. The common point of these algorithms, including also that proposed by Gasteiger and Jochum,[6] is the derivation of the SSSR from all cycles or other sets of basis rings. Plotkin[7] has developed one of the first algorithms for directly finding the SSSR. Other algorithms have been proposed by Bersohn,[8] Esack,[9] Zamora,[10] and Baumer *et al.*[11] for the same purpose.

All reported algorithms are based on two main methods, a walk through the connection table,[8-10,12,13] or graph manipulations.[4,5,11,14,15] Some theoretical aspects have been summarized by Lynch *et al.*[1] Our algorithm relies on the first approach, and requires as input data only a connection table, which is obtained from a routine for graphical acquisition of the target molecule. An example of the connection table is given in Table I for the molecule shown in Figure 1.

Table I. Connection Table for Molecule No. 1

atom no.	connection		
1	2	9	8
2	8	1	3
3	2	4	
4	5	3	
5	6	4	10
6	5	7	
7	6	8	
8	1	7	2
9	1	10	
10	9	5	

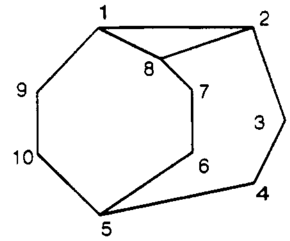


Figure 1. Molecule no. 1. The corresponding connection table is shown in Table I.

The SSSR found in a target molecule may be used to generate other types of ring sets, such as the extended set of smallest rings, defined and used by Lynch *et al.*,[17,18] or the set of all possible cycles in a given structure.

ALGORITHM

Before explaining the algorithm, we recall some basic terms and introduce several new definitions:

(a) A RING is represented as $R(n_1,n_2,...)$, where n_i are atom indices in a molecule. A SSSR is noted as $S(m_1,m_2,...)$, where m_i are the sizes of rings.

(b) Each atom is a NODE.

(c) The first selected NODE is a ROOT. A CLOSED PATH is a walk that starts and ends at a ROOT.

(d) The CONNECTIVITY of a node is defined as the number of its links with other nodes.

(e) A TREE is a structure derived from a root.

(f) If a node has only unit connectivity, it is called TERMINAL.

(g) A BLOCK is a group of atoms such that all links between them are involved in one or more rings. For instance molecule no. 2 of Figure 2 contains three blocks (A, B, and C).

(h) The REAL RING CONNECTIVITY of a node is the number of the links that are the edges of one or more rings in a block. It is denoted as N_m (m = 2, 3, 4). For example, the real ring connectivity for atom 8 in molecule no. 2 (see Figure 2) is only 2, not 3.

0095-2338/93/1633-0657$04.00/0 © 1993 American Chemical Society

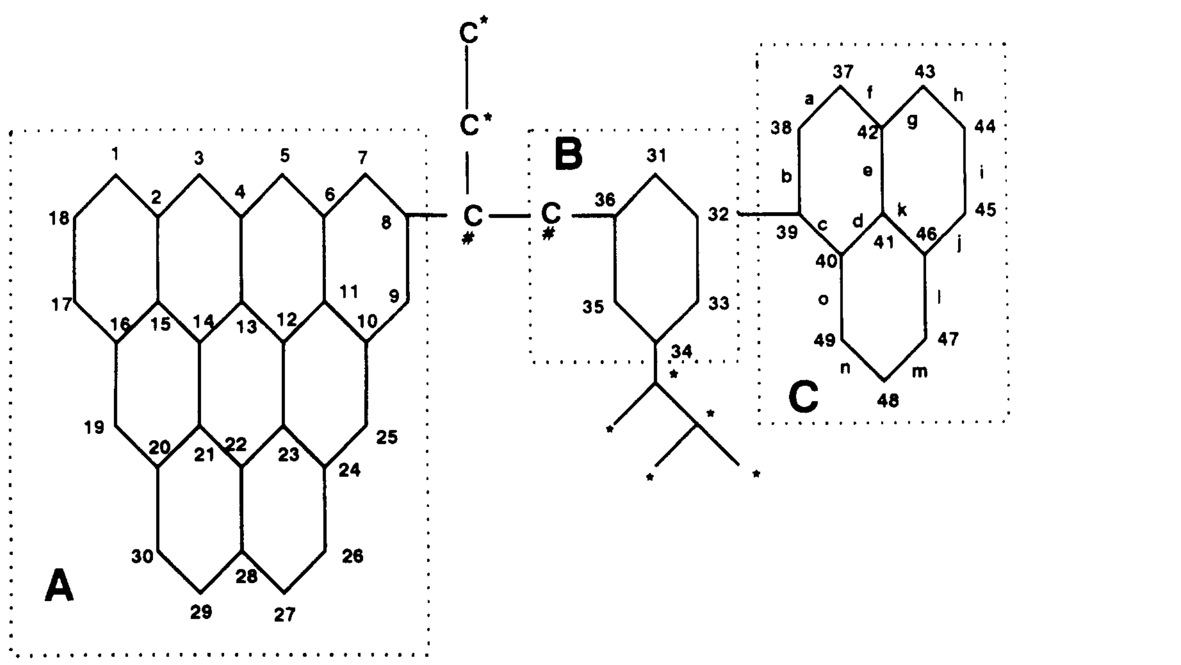


Figure 2. Example of separated blocks. This molecule (no. 2) can be divided into three blocks (A–C). For block C, four different categories can be distinguished. (1) All cycles: there are seven possible cycles, *R*(37,38,39,40,41,42), *R*(41,42,43,44,45,46), *R*(40,41,46,47,48,49), *R*(37,38,39,40,49,48,47,46,41,42), *R*(37,38,39,40,41,46,45,44,43,42), *R*(42,43,44,45,46,47,48,49,40,41), and *R*(37,38,39,40,49,48,47,46,45,44,43,42). (2) Set of simple cycles (β-rings): there are six β-rings, C1(a,b,c,d,e,f), C2(e,g,h,i,j,k), C3(d,k,l,m,n,o), C4(a,b,c,d,f,g,h,i,j,k), C5(a,b,c,e,f,k,l,m,n,o), and C6(d,e,g,h,i,j,l,m,n,o). (3) Fundamental basis set of rings: there are several possibilities for this category. Possibility 1°: C1(a,b,c,d,e,f), C2(a,b,c,d,k,j,i,h,g,f), and C3(d,o,n,m,l,k). Possibility 2°: C1(a,b,c,d,e,f), C2(g,h,i,j,k,e), and C3(o,d,k,l,m,n). Possibility 3°: ...... Etc. (4) SSSR: There is only one possible set, *S*(6,6,6) which contains three rings, *R*(37,38,39,40,41,42), *R*(41,42,43,44,45,46), and *R*(40,41,46,47,48,49). It is obvious that the possibility 2° of the fundamental basis set is the SSSR. This fundamental basis set is then considered as irreducible.

(i) An OPEN ACYCLIC NODE is an acyclic atom which is not located between two blocks, such as the atoms marked with an asterisk in molecule no. 2 of Figure 2. A CLOSED ACYCLIC NODE is an acyclic atom located between two blocks, as the atoms marked with the symbol # in molecule no. 2.

Our algorithm is an application to ring perception of the hill climbing searching and the least-cost searching techniques.[16] These techniques are commonly used in flight scheduling for instance to minimize the number of steps and the distance covered to join two cities by plane. They are claimed to be more efficient than usual, blind tree-search (either breadth-first or depth-first) methods since heuristic criteria are used. Formally, the hill-climbing technique chooses for its next step the node that appears to place it closest to the goal. It derives its name from the analogy of a hiker being lost in the dark halfway up a mountain. Assuming that the hiker's camp is at the top of the mountain, the hiker knows that each step upward is a step in the right direction. Least-cost searching is opposite of hill climbing. It takes the path of least effort.[16] In the heuristic used for our ring perception problem, one constraint is to minimize the number of node connections along a path (search for smallest rings). Another constraint is that a node with large real ring connectivity has greater likelihood to enter into a path. To obtain the SSSR rapidly, it is necessary to control the search in a correct direction (hill climbing) and to eliminate the path or nodes which can be certainly excluded in the procedure (least-cost). If we find a cycle from a root and it is the smallest ring, it is undoubtedly an element of the SSSR. The set of all of these smallest rings constitutes the SSSR. This routine is described in detail below.

We should emphasize that a recursive function (module), CYCSEARCH( ), used in the program, plays a primary role. This function allows one to remove all closed acyclic atoms, to separate the blocks, and to detect the paths from a root.

Step I. The procedure starts by removing all open acyclic atoms, beginning with terminal nodes (connectivity = 1). This is carried out by means of a recursive function (SUP_ACYC( )) (refer to the molecule no. 2 in Figure 2, all atoms marked with * are removed from the molecule in this step).

Step II. The recursive function (module) CYCSEARCH( ) removes all closed acyclic nodes (see molecule no. 2). The principle is relatively simple. Starting from a particular atom (root), for example atom 1 in molecule no. 2, if a closed path can be found, this atom surely is a ring member. However, the closed path found may or may not be a ring of the SSSR. This does not matter. The search is stopped as soon as the the first closed path is found. In contrast, a closed acyclic atom can never be found in a closed path (refer to the atoms marked with # in molecule no. 2). If *k* is the connectivity of a closed acyclic atom, this atom can be determined by running *k* times the recursive function CYCSEARCH( ). But in the case where an adjacent closed acyclic atom was previously removed from the molecule, the detection is simplified due to the decrease in the connectivity. Once such an atom is found, the program puts it in a separate table. After this step, all remaining atoms are nodes of at least one cycle.

Step III. The same recursive function CYCSEARCH( ), defined in step II, is used to separate blocks. As pointed out by Lynch *et al.*,[1] the separation of blocks gives increasingly dramatic improvements as the structure becomes more complex. For two atoms in the same block, two conditions will be satisfied during a complete topological search: (1) Each atom will be found in at least one closed path; (2) there is at least one crossing point, a common element for these atoms in their possible closed paths (but possibly with different locations in the two vectors describing the closed paths). In contrast, for two atoms of different blocks, there is never any

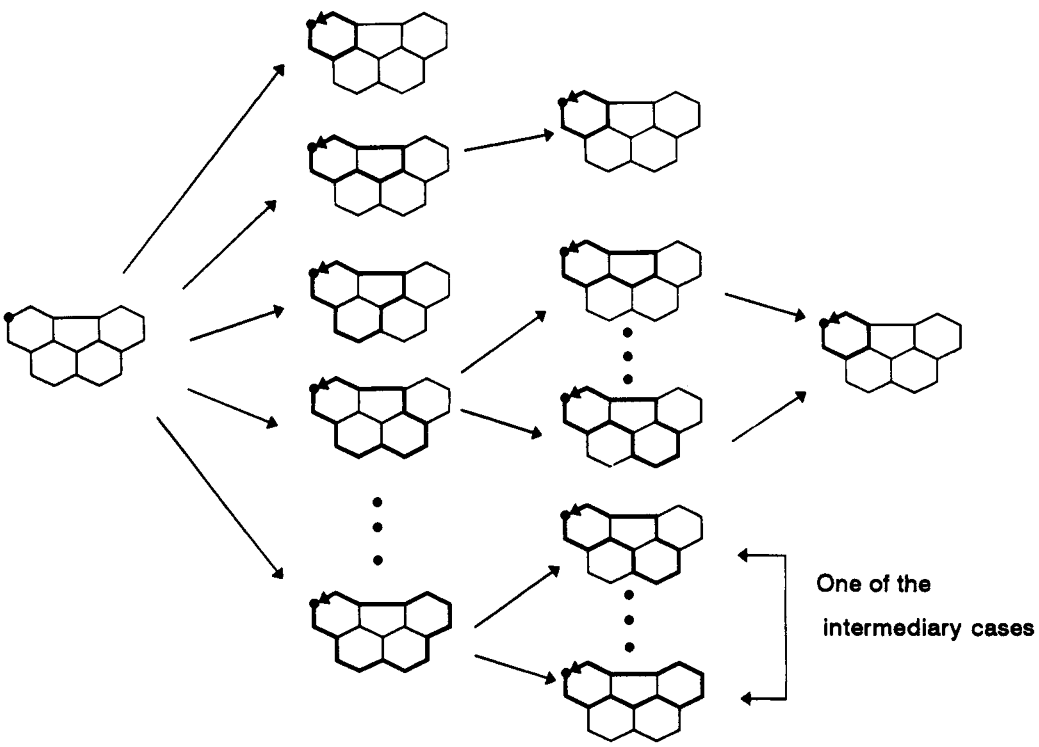


Figure 3. Mechanism of SSSR searching procedure.

crossing point in their possible paths. For example, it is impossible to find a crossing point for atom 31 and atom 1 in molecule no. 2. These two conditions are the basis for step III.

Step IV. The number of rings in a given block for the SSSR can be calculated with the following formula:

$$n_R = 1 + \frac{n(N_3)}{2} + n(N_4)$$

where the $n(N_3)$ and $n(N_4)$ terms are the number of nodes with real ring connectivities 3 and 4, respectively. The number of rings is in fact not needed for our algorithm but is used for control.

This step, the principal one, can be divided into substeps as follows.

(1) Choice of the Root Atom. In Zamora's algorithm,[10] the first ring found should be the central ring, that possesses the greatest connectivity for its atoms; the starting atom is that with the greatest connectivity. In our algorithm, the selection of the starting atom is not mandatory. However, to increase efficiency, a simple selection is performed. In contrast to the constraint of Zamora's algorithm, an unused atom with the smallest real ring connectivity will be selected as root; i.e., the atoms of connectivity $2(N_2)$ are *a priori* selected. If there is no atom with N_2 in the molecule, an atom with N_3 may be chosen.

(2) Search for the Smallest Ring Containing the Root. After the selection of the starting atom, the search begins by finding a closed path containing this root atom. An example is shown in Figure 3. From the root, marked by the bold point, the closed path is not unique. We cannot predict which path will be taken. We show several possible cases in the figure. The first path displayed is the best case where one pass gives the smallest ring. The last one, the worst case covers the whole molecule. Others are intermediary between these two extreme cases. In spite of the uncertainty in the search path, it is certain that a closed path can be found within the block.

The closed path found is stored in a vector. One of the smallest rings containing the root atom is included in this closed path. The next search step is confined to the zone enclosed by this vector. This path is scanned to find the first node with connectivity > 2, which must be in the zone. The next search takes another path within the limited sector. The repetition of this procedure leads to the smallest ring containing the starting atom. This ring is an *irreducible* closed path. Two advantages accrue from this procedure: (i) The number of nodes passed during the search generally decreases with each pass; (ii) the procedure can be written as a recursive function (or module).

(3) Elimination of the Reducible Atoms. Once a ring is found, the size of the block can be reduced by removing the atoms with connectivity N_2, according to the following criteria.

(i) If all N_2 atoms present are grouped together as a sequential string, these atoms can be removed from the block. For example, if the first ring found for molecule no. 3 (refer to Figure 4) is R(5*,1*,4,6,7,8*) or R(1*,5*,8*,7,6,4), all atoms of N_2 (marked with an asterisk) can be removed from the block. The residue will be as that shown in Figure 4.

(ii) If the N_2 atoms are not all adjacent in the ring vector, the segment containing most N_2 atoms is removed from the block. Remaining atoms of N_2 must be tested for another ring membership. The example shown in Figure 5 illustrates this operation. We assume that the rings found are R(5*,7,2,14*,15*,10,8) for molecule no. 4 and R(2*,3,5*,4,1*) for no. 5, respectively. In the first case (Figure 5a), the segment containing atoms 14 and 15 may be eliminated, leaving a residue as shown in the figure. Atom 5 is then tested. Of course, the result of the test shows that this node does not belong to another cycle. It can be also removed from the block. In the second case (Figure 5b) after the removal of the segment containing atoms 2 and 1, a test shows that atom 5 is still in another ring system. It must be maintained.

This procedure seems to be complicated but is easy to implement. Comparing this test with step II, it is evident that the module used in step II can be called for this purpose. No more program is needed.

(iii) In the case where we have same number of N_2 atoms for all segments, a supplementary treatment is necessary.

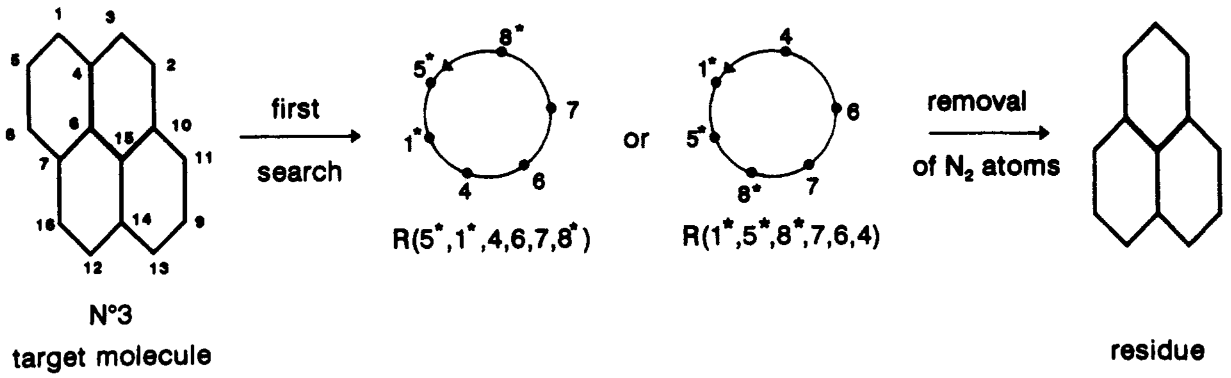


Figure 4. First case of removing the N_2 atoms.

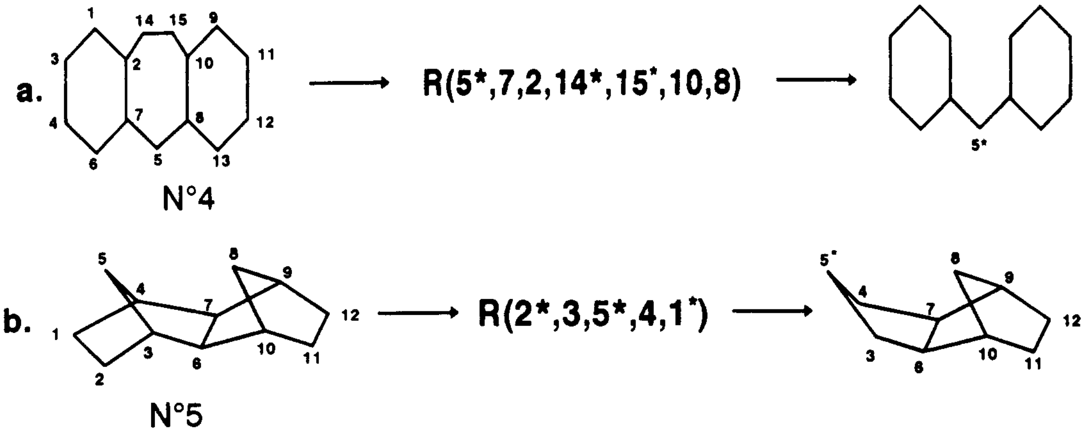


Figure 5. Second case of removing the N_2 atoms.

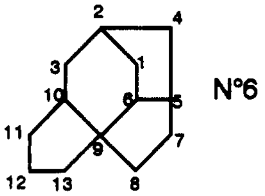


Figure 6. Molecule with a same number of N_2 atoms for all segments. We thank the reviewer who proposed this example to us.

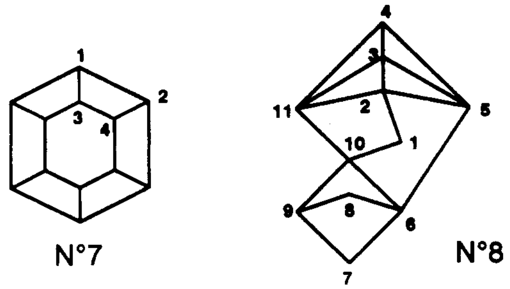


Figure 7. Molecule without real ring connectivity of N_2 (no. 7) and molecule having several possible SSSRs (no. 8).

Considering molecule no. 6, shown in Figure 6, as an example, atom 1 is selected as root. The first ring found may be R_1(1,2,3,10,9,6) or R_2(1,2,4,5,6). We encounter a delicate situation when we pass to the N_2 atom removal step. There are two segments with the same number of N_2 atoms (here equal to 1). Atom 1 must be kept for the next search step. This problem is solved by a comparison test, described as follows.

When the numbers of N_2 atoms are equal for all segments, the N_2 atoms appearing in the segment which contains the actual root are provisionally removed. One of the N_2 atoms belonging to an other segment is selected as the root atom. The smallest ring containing this new root is stored in a buffer table. The N_2 atoms in this segment are removed, and the N_2 atoms previously removed are restored. The same procedure repeats until all segments are considered. Only the smallest ring is recorded after a comparison of all temporary rings. In our example, if R_1 is initially found, atom 1 is provisionally removed. Atom 3 is selected as a new root. A seven membered ring R(3,2,4,5,6,9,10) is found as the smallest one containing the root atom. This ring is stored in a separate table. The next step is restoring atom 1 and removing the atom 3. We find another five membered ring R(1,2,4,5,6). By comparison with the previous one, this new ring is substituted to the seven membered ring in the buffer table. Because there are only two segments, the search for this part is finished and the latter ring is recorded as a member of the SSSR.

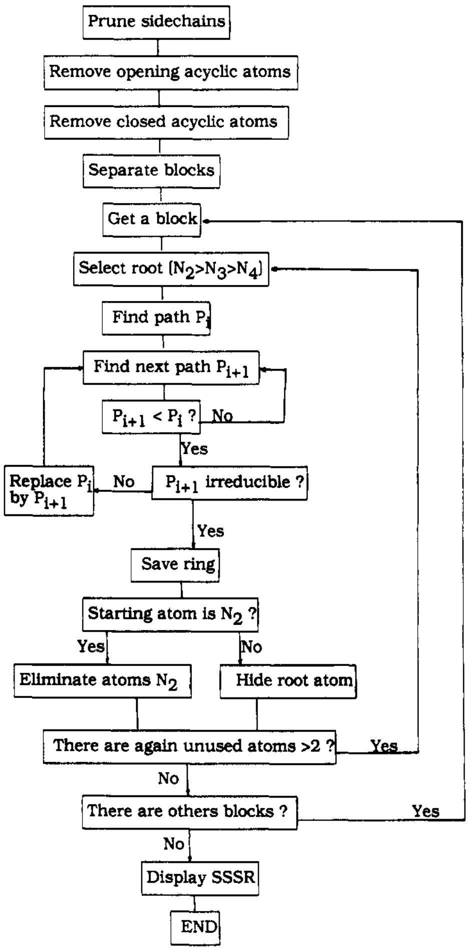
In fact, procedure iii may be combined with procedure ii to ensure that the rings found are the smallest ones.

After removing the N_2 atoms, the neighboring atoms lose one degree for their real ring connectivity. The procedure repeats until all atoms are eliminated from the block.

(4) Hiding the Root Atom of N_3 or N_4 Connectivity. In some cases, such as molecule no. 7 (Figure 7), there is no N_2 atom in a block, so we cannot remove N_2 atoms, as previously detailed. Each atom belongs at least to two rings. This problem is treated by hiding the starting atom after finding its smallest ring. One of the neighboring atoms attached to the hidden atom loses one degree of connectivity. Consider molecule no. 7 as an example. Once the ring R(1,2,4,3) is found, the starting atom, here atom 1, is hidden. The connectivity of atom 2 (or atom 3) decreases by 1. For the next ring search, atom 2 (or atom 3) will be selected owing to its lower connectivity relative to other nodes. The hidden atom (1) will be restored later, for further searching.

Step IV is iterated to find all smallest rings in the block. They are placed in a set. The same procedure is applied to the next block, etc., until all blocks are treated.

Scheme I. Flow Chart of the Routine FINDCYC( )



As a consequence of the algorithm, each ring found is ensured to be the smallest ring (or one of the smallest cycles with equivalent size) containing the starting atom. The elimination technique guarantees that no ring is doubly found and stored. It is thus certain that the set of these rings is the SSSR, and the SSSR is found directly without any accessory processes.

If there are several possible SSSRs in a target molecule, the SSSR found will be arbitrary. The algorithm cannot predict which SSSR will be found. For example, in target molecule no. 8 (see Figure 7), $n(N_3)$ = 2 (atoms 4 and 9) and $n(N_4)$ = 6 (atoms 2, 3, 5, 6, 10, and 11). The number of rings is

$$n_R = 1 + n(N_3)/2 + n(N_4) = 8$$

The SSSR found should be S(3,3,3,3,4,4,4,5). For the three membered rings, there is no ambiguity. They are R(2,3,5), R(3,4,5), R(2,3,11), and R(3,4,11). But for the four membered rings, there are two possibilities: (i) R(1,2,10,11), R(6,8,9,10), and R(6,7,8,9); or (ii) R(1,2,10,11), R(6,7,9,10), and R(6,8,9,10). Similarly, the five membered ring may be one of the following rings: R(5,6,10,11,2), R(5,6,10,11,3), or R(5,6,10,11,4). The SSSR found is entirely dependent upon the structure of the connection table, obtained from the graphics acquisition procedure. But it is certain that whatever the number of possible SSSRs in a target molecule, the program will find one of them.

Scheme I shows the flow chart of the main FINDCYC procedure.

EXAMPLES

Molecules 1–8 were treated with FINDCYC. The results obtained are gathered in Table II. The results for some complex ring systems (9–12) (Figure 8) are also shown in Table II.

Table II. Examples of SSSR Calculation

molecule no.	SSSR	molecule no.	SSSR
1	S(3,7,7)	7	S(4,4,4,4,4,4,6)
2	S(6,6,6,6,6,6,6,6,6,6,6,6,6,6)	8	S(3,3,3,3,4,4,4,5)
3	S(6,6,6,6)	9	S(5,5,6,6,6)
4	S(6,6,7)	10	S(5,5,5,5,6,6)
5	S(5,5,5,5)	11	S(3,4,4,5,5,6,6,7)
6	S(5,5,5,6)	12	S(4,4,4,5,5,5,5,5,5)

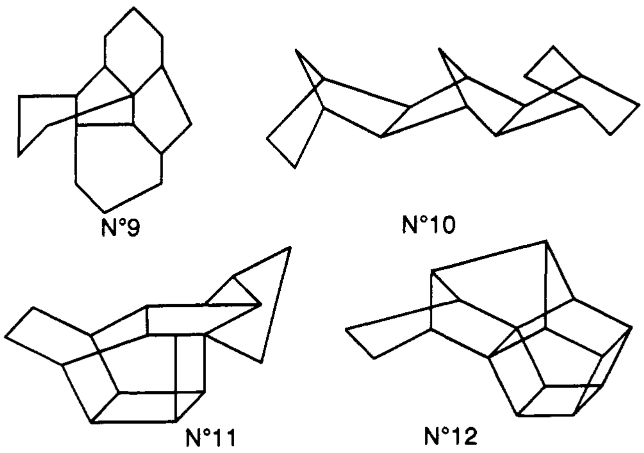


Figure 8. Some molecules with more complexity in their structure.

IMPLEMENTATION

The program is written in C++ under MS-DOS. It is implemented on an IBM-PC using Borland's Turbo C++ compiler. The SSSR rings are stored using a data structure in binary form. The source code is portable to the UNIX operating system without modification, making possible an implementation on a workstation.

CONCLUSION

A relatively simple algorithm for ring perception supplies a tool for directly finding the SSSR from a connection table. In some previous algorithms, rings have to be divided in various classes (three classes for Zamora's algorithm,[10] two classes for Baumer *et al.*[11]). We do not need such a classification for different ring types. The input data are a simple connection table, obtained from a graphics acquisition routine. No matrix is introduced into the calculation.

This algorithm saves memory space by the fact that every stored cycle is an element of the SSSR. It is not necessary to store all rings, to be used later for deriving the SSSR, as implemented by other authors.

REFERENCES AND NOTES

(1) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Review of ring perception algorithms for chemical graphs. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 172–187.

(2) Corey, E. J.; Petersson, G. A. An algorithm for machine perception of synthetically significant rings in complex cyclic organic structures. *J. Am. Chem. Soc.* **1972**, *94*, 460–465.

(3) Wipke, W. T.; Dyott, T. Use of ring assemblies in a ring perception algorithm. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 140–144.

(4) Welch, J. A mechanical analysis of the cyclic structure of undirected linear graphs. *J. ACM* **1966**, *13*, 205–210.

(5) Gibbs, N. A cycle generation algorithm for finite undirected linear graphs. *J. ACM* **1969**, *16*, 564–568.

(6) Gasteiger, J.; Jochum, C. An algorithm for the perception of synthetically important rings. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 43–48.

(7) Plotkin, M. Mathematical basis of ring-finding algorithms in CIDS. *J. Chem. Doc.* **1971**, *11*, 60–63.

(8) Bershon, M. An algorithm for finding the synthetically important rings of a molecule. *J. Chem. Soc., Perkin Trans. 1* **1973**, 1239–1241.

(9) Esack, A. A procedure for rapid recognition of the rings of a molecule. *J. Chem. Soc., Perkin Trans. 1* **1975**, 1120–1124.

(10) Zamora, A. An algorithm for finding the smallest set of smallest rings. *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 40–43.

(11) Baumer, L.; Sala, G.; Sello, G. Ring perception in organic structures: a new algorithm for finding SSSR. *Comput. Chem.* **1991**, *15* (4), 293–299.

(12) Tiernan, J. An efficient search algorithm to find the elementary circuits of a graph. *Commun. ACM* **1970**, *13*, 722–726.

(13) Corey, E. J.; Wipke, W. T.; Cramer, R. D.; Howe, W. J. Techniques for perception by a computer of synthetically significant structural features in complex molecules. *J. Am. Chem. Soc.* **1972**, *94*, 431–439.

(14) Fujita, S. Logical perception of ring opening, ring closure, and rearrangement reaction based on imaginary transition structures. Selection of the essential set of essential rings (ESER). *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 1–9.

(15) Fujita, S. A new algorithm for selection of synthetically important rings. The essential set of essential rings for organic structures. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 78–82.

(16) Schildt, H. In *Turbo C: The complete reference*; Borland, Osborne, Ed.; McGraw-Hill: New York, 1988.

(17) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Theoretical aspects of ring perception and development of extended set of smallest rings concept. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 187–206.

(18) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Computer storage and retrieval of generic chemical structures in patents. 9. An algorithm to find the extended set of smallest rings in structurally explicit generics. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 207–214.