An Algorithm for Finding the Smallest Set of Smallest Rings

ANTONIO ZAMORA

Chemical Abstracts Service, The Ohio State University, Columbus, Ohio 43210

Received October 16, 1975

This paper describes an algorithm which finds the smallest set of smallest rings of a ring system without the necessity of finding all rings in the ring system. The algorithm first finds the smallest rings in which unused atoms occur and then progresses to find the smallest rings in which unused edges and faces occur until the smallest set of rings required to describe the complete ring system is found. The algorithm converges quickly because the lengths of the paths that need to be scanned to discover each new ring decrease when a smaller ring is found.

INTRODUCTION

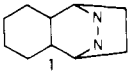
Chemical information systems have traditionally handled cyclic chemical compounds on the basis of their structural characteristics, such as ring sizes and atom population of rings. Since a ring system can have many different sets of rings which describe it equally well, it is customary to characterize a ring by the smallest set of smallest rings (SSSR). "The Ring Index"[1] and notation systems such as that originated by Wiswesser[2] make use of the smallest set of smallest rings.

The minimum number of rings required to describe a ring system is easy to obtain. It corresponds to the number of cuts required to convert the ring system into a single open-chain structure and is given by the equation:

$$\text{RINGS} = \text{EDGES} - \text{ATOMS} + 1$$

where the number of edges is the number of distinct atom-to-atom connections.

For a ring system with indistinguishable nodes, the atom population of the rings is dependent only on the ring sizes. However, for heterocyclic ring systems which have distinct nodes, the atom population can depend on the subset of rings selected even when the SSSR is selected. Thus the atom population or elemental ring analysis for **1** can be either



1

C_4N_2–C_4N_2–C_6 or C_4N_2–C_6–C_6. A chemical information system may choose between these two elemental ring analyses based on the best locant numbering for the corresponding chemical name or linear notation.

Welch[3] and Gibbs[4] have provided methods for finding all the rings of a ring system by starting from an arbitrary set of fundamental cycles, and Gotlieb and Corneil,[5] Paton,[6] and Tiernan[7] have described algorithms for finding fundamental cycles.

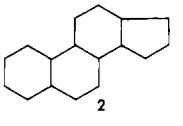
Obtaining the SSSR, as shown by Plotkin,[8] is a conceptually simple procedure. In essence, all the rings of a given structure are sorted by ring size. The smallest ring is always assigned to the SSSR. Rings other than the first are examined in ascending sequence by ring size and are added to the SSSR only if they are linearly independent from any rings previously added to the SSSR. If the rings are subjected to a secondary ordering by elemental composition, in addition to size, it is also possible to select a preferred elemental ring analysis for the structure. In practice this procedure is quite time consuming even with today's computers, and for this reason alternative procedures are still being sought.

This paper presents a path-tracing algorithm for finding the SSSR; the algorithm can easily be modified to also give an elemental ring analysis. Where more than one elemental ring analysis exists for a ring system, the one selected will depend on the parameters coded in the algorithm. This algorithm is currently used at Chemical Abstracts Service to edit the output of a program which generates Wiswesser Line Notations.[9] Although the algorithm has some limitations, the occurrence of ring systems for which these limitations are evident is rare.

CLASSES OF RING SYSTEMS

For the purpose of this study, ring systems are subdivided into three types:

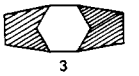
Type I. Ring systems for which no subset of the smallest rings contains all the atoms of the ring system (**2**).



2

SSSR : 5,6,6,6

Type II. Ring systems for which all the atoms but not all the edges of the ring system are contained by a subset of the smallest rings, for example, in **3** the two shaded rings contain all the atoms but do not contain two of the edges.



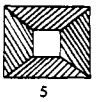
3

SSSR : 5,5,6

Type III. Ring systems for which all the atoms and all the edges of the ring system are contained by a subset of the smallest rings (**4**, **5**, and **6**).

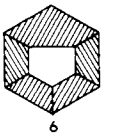
4

SSSR : 3,3,3,3,4



5

SSSR : 4,4,4,4,4



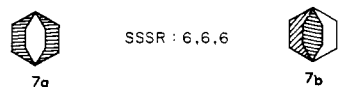
6

SSSR : 4,4,4,4,5,5

SELECTION OF SSSR. PHASE 1

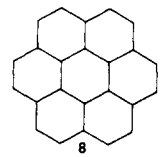
The SSSR can be found for ring systems of type I by randomly selecting an unused atom and finding the smallest ring which contains that atom. The atoms comprising the smallest ring found are marked used, and the process is repeated until all atoms are marked used.

This procedure will also give the SSSR for certain ring systems that are not of type I but can be reduced to type I by proper selection of the starting atom and by selection of rings with the greatest number of used atoms when there is a choice between rings of the same size. The following examples illustrate how, by judicious rather than random selection of the rings and the starting atoms, certain ring systems can be reduced to type I. If the shaded rings in **7a** were found



first, all the atoms would have been marked used before all the rings had been found. However, by selecting as the second ring a ring that has the greatest number of used atoms (still starting from an unused atom), two unused atoms will remain for the last ring as shown in **7b**.

For a structure such as **8** the choice of a starting atom, as



SSSR : 6,6,6,6,6,6,6

well as the first ring, can make a significant difference; if the central ring is isolated first, the peripheral rings can be found as before, since each one will have two atoms not shared by any other ring.

The central ring can be found by use of a property of the graph called connectivity. For the purposes of this algorithm connectivity is defined as follows:

Let K_i = 1, 8, or 64 depending on whether atom i has 2, 3, or 4 attachments, respectively. (Four is an arbitrary limit within this algorithm.)

Let L_i be the sum of the K values of the atoms to which atom i is attached; the connectivity of atom i is then $C_i = 64(K_i) + L_i$.

Thus, the connectivity of an atom depends primarily on its number of attachments, but is also influenced by the environment of the attached atoms. Note that for **8** the atoms of the central ring have the greatest connectivity.

The SSSR for **8** can be found by selecting as the starting atom the unused atom with the greatest connectivity and then selecting the smallest ring that contains the atom. If more than one such ring exists, the ring containing the atoms with the greatest connectivity sum is selected. The atoms of the selected ring are marked used, and the process is repeated until all atoms are marked used.

The procedure given above (PHASE 1 of the SSSR algorithm) gives the SSSR for type I ring systems and those which can be reduced to type I. Ring systems for which PHASE 1 does not give all the expected rings (known from the equation: RINGS = EDGES − ATOMS + 1) are subsequently examined by PHASE 2 and, if necessary, by PHASE 3 of the algorithm to produce the SSSR. The SSSR algorithm, then, consists of three phases applied in succession and terminating as soon as the SSSR is found.

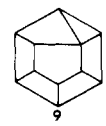
SELECTION OF SSSR. PHASE 2

The SSSR for ring systems of type II can be found by means of a usage counter which records the number of times each edge has been used as each ring is selected. If any edges remain unused after PHASE 1, the smallest ring in which an unused edge occurs is selected and the usage counters of the edges in the selected ring are incremented. The process is repeated until all edges have been used. When there is a choice between rings of the same size, the ring with the greatest number of used edges is selected.

SELECTION OF SSSR. PHASE 3

If all the rings have not been found after all the atoms and edges have been used, we proceed to find all unused faces of the ring system and choose as many of the smallest faces as are required to fully describe the ring system. An unused face is defined here as a ring consisting exclusively of atoms having at least three attachments. Furthermore, only one of the edges of the ring is allowed to have been used more than once; all other edges must have a usage count of one.

It should be noted that on occasion the algorithm may need to choose between more than one ring during PHASE 3. For example, in **9** the choice of the final ring will be between the



SSSR : 3,4,4,4,4,4,5

internal pentagon and the peripheral hexagon. In **6**, however, only the pentagon is available as a final choice since the peripheral hexagon does not meet the definition of an unused face.

THE ALGORITHM

A ring-finding algorithm is presented here followed by the SSSR algorithm which incorporates all the features described above. Rings are found by tracing paths that eventually terminate at the starting atom. All such paths which do not exceed the number of atoms of the smallest ring found previously are tried. Thus, each time a smaller ring is found, the length of the paths that need to be scanned also decreases.

DEFINITIONS

FIRST	The atom selected as the first atom of the path; the ring-finding algorithm will select the smallest ring containing this atom
SM	Size of the smallest ring found
SIZE	Number of atoms in the current path
$A[j]$	Number of attachments to atom j
GRAPH[i,j]	Graph description where i = 1, 2, ..., $A[j]$ defines the atoms to which atom j is attached and j = 1, ..., N where N is the number of atoms in the graph.
$X[j]$	The attachment to the jth atom currently being examined ($1 \leq X[j] \leq A[j]$)
$P[k]$	List of atoms in the current path, k = 1, ..., SIZE
USE[j]	Usage indicator USE[j] = 1 if atom j is in $P[k]$; otherwise it is zero
BEST[k]	List of atoms in the smallest ring k = 1, ..., SM

RING-FINDING ALGORITHM

```
/* INITIALIZATION */
    SM ← N + 1
    X ← 0
    SIZE ← 1
    P[SIZE] ← FIRST
    CURRENT ← FIRST
    USE[FIRST] ← 1
/* PATH EXPLORATION */
  ALPHA: X[CURRENT] ← X[CURRENT] + 1
    IF X[CURRENT] > A[CURRENT] GO TO SCAN
    ATTACHED ← GRAPH[X[CURRENT],CUR-
    RENT]
    IF SIZE > 1 & ATTACHED = P[SIZE-1] GO TO
ALPHA
    IF USE[ATTACHED] = 1 GO TO BETA
/* PATH EXTENSION */
    SIZE ← SIZE + 1
    P[SIZE] ← ATTACHED
    USE[ATTACHED] ← 1
    CURRENT ← ATTACHED
```

```
    IF SIZE > SM GO TO SCAN
    GO TO ALPHA
/* RING CONFIRMATION */
  BETA: IF ATTACHED ≠ FIRST GO TO ALPHA
/* A RING HAS BEEN FOUND */
    IF SIZE < SM GO TO BETTER
/* THE RINGS ARE OF EQUAL SIZE -- INSERT
  CRITERIA FOR CHOICE AT THIS POINT.
    IF THE RING IS NOT BETTER GO TO SCAN */
BETTER: SM ← SIZE
          BEST ← P
/* SCAN BACKWARDS ALONG PATH */
  SCAN: X[CURRENT] ← 0
    USE[CURRENT] ← 0
    SIZE ← SIZE - 1
    CURRENT ← P[SIZE]
    IF SIZE = 1 & X[CURRENT] + 1 ≥ A[CURRENT]
    GO TO DONE
    GO TO ALPHA
DONE:
  /* THE SMALLEST RING IS IN BEST[1:SM] */
  END
```

To find the smallest ring with a specific edge, the initialization of the ring-finding algorithm is as follows:

```
SM ← N + 1
X ← 0
P[1] ← FIRST
P[2] ← SECOND
X[FIRST] ← A[FIRST]
SIZE ← 2
USE[FIRST] ← 1
USE[SECOND] ← 1
CURRENT ← SECOND
GO TO ALPHA
```

The variables FIRST and SECOND in this case represent the two nodes joined by the edge.

SSSR ALGORITHM DESCRIPTION

PHASE 1

1. Select the unused atom that has the greatest connectivity; if there are no unused atoms go to PHASE 2.
2. Apply the ring-finding algorithm with the following modifications:
   a. During path extension add the connectivities of the atoms in the path and keep a count of the number of used atoms in the path. Also record the number and kind of non-carbon atoms present if an elemental ring analysis is desired.
   b. If there is a choice between rings of equal size, choose the ring that has (i) the greatest number of used atoms; (ii) the greatest number of used edges; (iii) the greatest connectivity sum; (iv) the preferred elemental analysis.
   c. When scanning backwards along the path, decrement the connectivity sum and restore other ring selection variables altered during path extension.
3. Mark the atoms of the selected ring "USED."
4. Increment the usage count of each edge of the ring by one.
5. Go to 1.

PHASE 2

6. Mark all the atoms "NOT USED."
7. If the number of rings found is equal to the smallest set of rings, go to END.
8. Select an unused edge, if no unused edges remain, go to PHASE 3.
9. Apply the ring-finding algorithm with the same modifications as PHASE 1.
10. Mark the atoms of the selected ring "USED."
11. Increment the usage count of each edge of the ring by 1.
12. Go to 7.

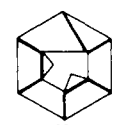
PHASE 3

13. Select an edge with a usage count of one. If none are present go to 20.
14. Set the usage count of the selected edge equal to two.
15. Apply the ring-finding algorithm with the following modifications: path extension is done only if both atoms of the edge have at least three attachments and if the number of edges in the path with usage count of two or greater does not exceed one.
16. If a ring is not found, go to 13.
17. If a ring is found, save the ring and its associated elemental analysis.
18. Increment the usage counts of the edges of the ring by one.
19. Go to 13.
20. Sort the rings found by PHASE 3 in increasing order by size and in preferred elemental analysis sequence.
21. Select rings from the sorted set, starting from the smallest ring, until the total number of rings selected equals the smallest set of rings.

END. End of SSSR algorithm.

ALGORITHM LIMITATIONS

An unused face has been defined as a ring consisting of atoms having at least three attachments with no more than one edge of the ring exceeding a usage count of one. This definition takes advantage of the fact that PHASE 1 and PHASE 2 of the algorithm will always maximize the usage counts of the edges when a choice of rings is possible. Thus, it is assumed in PHASE 3 that nonelementary rings or previously discovered rings would be found by using more than one edge with usage count of two or greater. This assumption is not always correct and leads to two types of errors. The failures appear to be restricted to situations where all the edges of the final ring are contained in several smaller rings and several edges of the final ring are used more than once. Structure **10** shows a case where

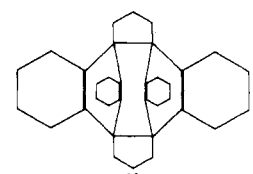


10

SSSR : 3,3,3,4,4,4,4,4,5

the final ring found by PHASE 3 would be a hexagon rather than a pentagon. The pentagon in this case has two edges with usage counts of two which disqualify it from being an "unused face". Edges that have been used twice are darkened in **10**. This is a theoretical example; no real case has been encountered.

The next example (**11**) illustrates a case where rather than



11

SSSR : 5,5,6,6,6,6,6,8

finding a ring of the wrong size, PHASE 3 fails to find the final eight-atom ring. Of the four distinct eight-atom rings present in the structure, all of them have two edges with usage count of two. The algorithm as programmed indicates this situation.

It would be possible to extend the algorithm to find the final rings for most cases when PHASE 3 fails by obtaining a set of fundamental rings for atoms of the structure having three or more attachments and then applying the procedure outlined

in the Introduction. The total number of rings generated in this way would generally be far fewer than if all the atoms of the structure were included; this reduction is possible because the first two phases of the algorithm create a partial SSSR which could be used to select the missing rings.

CONCLUSION

This paper has presented a fairly comprehensive algorithm for finding the smallest set of smallest rings. The speed with which the smallest rings are found is dependent on the sequence in which the paths are explored. This, in turn, depends on the way in which the atoms are numbered when input to the algorithm. Although the algorithm can be shown to fail in some cases, the ring systems for which it fails constitute a very minute portion of those which are chemically possible.

The algorithm was programmed in PL/1 and required 258 statements. It should be noticed that the basic procedure for finding the smallest set of smallest rings is independent of the technique used to implement the ring-finding algorithm. Although the ring-finding algorithm illustrated in this paper uses a path-tracing technique, other techniques such as growing a tree from the selected atom or atoms might offer advantages in particular situations.

ACKNOWLEDGMENT

The author thanks T. Ebe and J. Mockus for their encouragement and stimulating discussions.

REFERENCES AND NOTES

(1) A. M. Patterson, L. T. Capell, and D. F. Walker, "The Ring Index", 2nd ed, American Chemical Society, Washington, D.C., 1960.
(2) E. G. Smith, "The Wiswesser Line-Formula Chemical Notation", McGraw-Hill, New York, N.Y., 1968.
(3) J. T. Welch, Jr., "A Mechanical Analysis of the Cyclic Structure of Undirected Linear Graphs", *J. Assoc. Comput. Mach.*, **13**, 205–10 (1966).
(4) N. E. Gibbs, "A Cycle Generation Algorithm for Finite Undirected Linear Graphs", *J. Assoc. Comput. Mach.*, **16**, 564–8 (1969).
(5) C. C. Gotlieb and D. G. Corneil, "Algorithms for Finding a Fundamental Set of Cycles for an Undirected Linear Graph", *Commun. Assoc. Comput. Mach.*, **10**, 780–3 (1967).
(6) K. Paton, "An Algorithm for Finding a Fundamental Set of Cycles of a Graph", *Commun. Assoc. Comput. Mach.*, **12**, 514–8 (1969).
(7) J. C. Tiernan, "An Efficient Search Algorithm to Find the Elementary Circuits of a Graph", *Commun. Assoc. Comput. Mach.*, **13**, 722–726 (1970).
(8) M. Plotkin, "Mathematical Basis of Ring-Finding Algorithms at CIDS", *J. Chem. Doc.*, **11**, 60–63 (1971).
(9) A. Zamora and T. Ebe, "PATHFINDER II. A Computer Program That Generates Wiswesser Line Notations for Complex Polycyclic Structures", *J. Chem. Inf. Comput. Sci.*, preceeding paper in this issue.

Principle for Exhaustive Enumeration of Unique Structures Consistent with Structural Information

YOSHIHIRO KUDO and SHIN-ICHI SASAKI*
Miyagi University of Education, Sendai, 980 Japan

Received March 26, 1975

Unique structures consistent with structural information are enumerated by means of the "connectivity stack", the proper situation to provide an effective examination of the correct estimation of each structure, complete or even under construction, as one of the members of the "informational homologues". Both cyclic and acyclic structures are treated.

We have developed an integrated system for structure elucidation of organic compounds,[1-4] and called it CHEMICS.[5] The generic acronym CHEMICS stands for Combined Handling of Elucidation Methods for Interpretable Chemical Structures. It is a system for deducing all logically valid structures,[6,7] acyclic and cyclic, on the basis of previously settled propositions according to input information concerned with the structure of a given compound. Each logically valid structure is defined as an *informational homologue*[5,8] of provided structural information. If the information consists of only a molecular formula, the informational homologues are identical with structural isomers, whose members may even exceed millions.[9] Their composition depends on only the nature of the provided information; that is, the richer the information, the fewer informational homologues there are. In order to enumerate them not only completely and uniquely but also as quickly as possible,[16] a new principle of enumeration has been devised and has yielded many results for CHEMICS,[1-4] though most are not published in the literature. It was recently known that the principle in the heuristic DENDRAL[10,11] is very similar to ours because of mathematical permutation, though the object and order of application are different from each other. Mathematical permutation is one of the best ways for exhaustive enumeration, but really has practical value when hopeless branches of a logical tree are eliminated as early as possible. Balaban's report[12] directly stimulated publication of the original principle of our enumeration methods.

REPRESENTATION OF THE STRUCTURES

The enumeration part of CHEMICS combines static features with dynamic ones. The former is to carry out correct enumeration and the latter is to decrease execution time. How to represent structures goes along with both features.

Component and Segment. Most chemical systems, e.g., DENDRAL,[11] CAS/Morgan,[13] IUPAC/Dyson,[14] WLN,[15] represent a structure with canonical connectivities and after this with segments under constraint of hierarchical orders, in their own peculiar ways. On the other hand, CHEMICS considers segments in a hierarchical order by their parent components first and secondly constructs a suitable connectivity representation according to the order. The two concepts, component and segment, correspond to chemical element and atom in general chemistry, respectively. That is, the component is a logical division of partial structures, and the segment is an entity with the component as property. After setting the components, each part of a whole molecule is always specified with exactly one component. Two conditions, (1) and (2), define the concept of the component, C_i:

$$\bigcup_i C_i = \text{all whole structures} \quad (1)$$

$$C_i \cap C_j = 0 \quad (i \neq j) \quad (2)$$

There are many possible ways to set up components under the two conditions. There is no natural component set and a